

Unix System Administrators Guide



Computron Software, LLC.
301 Route 17 North
Rutherford, NJ 07070
+1-201-372-6100
www.ComputronSoftware.com

Revision History

First Edition		April 1994
Second Edition		December 1996
Third Edition	Version 5.0	December 1998
Fourth Edition	Version 5.0	September 1999
Fifth Edition	Version 7.0	August 2002
Sixth Edition	Version 8w	June 2003
Seventh Edition	Version 10.0	November 2006
Eighth Edition	Version 11.0	December 2008
Ninth Edition	Version 12.0	July 2010

Disclaimer of Warranties and Limitation of Liabilities

The information in this document is valuable and confidential proprietary information belonging solely to Computron Software, LLC. Nothing contained herein modifies or alters in any way the standard terms and conditions of Computron's license agreement by which this software package was acquired, nor increases in any way Computron's liability to the customer. Without limiting the generality of the foregoing, in no event shall Computron be liable for incidental, consequential, or other indirect damage in connection with or arising from the use of the software package, the accompanying documentation, or any related material. Information herein may change without notice and Computron assumes no responsibility for any errors that may appear in this document.

This software documentation is for the internal use of Computron personnel and licensed customers only. It is not to be used, disclosed, disseminated or reproduced for any other purpose without prior written consent by Computron Software, LLC.

© Copyright 1994 - 2010, Computron Software, LLC. All rights reserved. This copyright notice is a precaution against inadvertent publication and does not constitute an acknowledgment that publication has occurred nor does it imply any waiver of confidentiality or proprietary rights.

NOTE: This document may contain "revision bars" located in the margin to the right of revised paragraphs/sections. Revision lines indicate changes to the documentation that pertain to product functionality. An example is at the right of this paragraph.

Sales/Support Locations

United States

Computron Software, LLC.
160 Chubb Ave
Lyndhurst, New Jersey 07071
+1-201-372-6100
+1-800-421-3136
www.computronsoftware.com

Europe

Computron Software Ltd.
88 Kingsway
London
WC2B 6AA
United Kingdom
+44 (0) 20 7190 1619

Pacific

Computron Software Australia Pty. Ltd.
Level 6, 77 Berry Street
North Sydney 2060
NSW Australia
+61 2-9929-8925

Computron Software Australia Pty. Ltd.
2nd Floor, 541 Blackburn Road
Mt Waverly 3149
VIC Australia
+61-3-9005-4910

South Africa

Computron (a division of Ctronsoft SA (Pty) Ltd)
8 Sycamore Road
Zwartkop
Centurion
Pretoria, South Africa
0157
+27 12 663 5221

Asia

Computron Software Pte. Ltd.
1 Phillip Street #12-02
Singapore 048692
(65) 6536-7808

System Administrators Guide – Unix

Table of Contents

Preface

Chapter 1—Universal Utilities

Overview: Setting Up Computron Users	1.1-1
User File Maintenance	1.2-1
User File Inquiry/List	1.3-1
Menu File Maintenance.....	1.4-1
Menu File Print.....	1.5-1
Database Map File Maintenance	1.6-1
Global Variable Maintenance.....	1.7-1
Global Variable Maintenance Inquiry/List.....	1.8-1
EPIC In UV	1.9-1
Soft Screen Maintenance.....	1.10-1
Capture Log File to Print.....	1.11-1
Lockout Utility (CSUNLOCK)	1.12-1
Lockout Inquiry	1.13-1
Lockout Delete by User.....	1.14-1
System Access Inquiry / List.....	1.15-1
Print Audit File	1.16-1

Chapter 2—Security Subsystem

Security Subsystem	2.1-1
Ownership Code Maintenance.....	2.2-1
Ownership Code Inquiry/List.....	2.3-1
User Profile Maintenance.....	2.4-1
User Profile Inquiry/List.....	2.5-1

Chapter 3—Job Processing

Job Processing - An Overview.....	3.1-1
Job Scheduler Code List Maintenance Functions.....	3.2-1
Job Scheduler Directory Maintenance	3.3-1
Job Scheduler Directory Inquiry/List	3.4-1
User Code Maintenance	3.5-1
User Distribution List Maintenance	3.6-1
User Distribution List Inquiry/List	3.7-1
Job Processor.....	3.8-1
Audit Processor	3.9-1

Return Code Inquiry/List	3.10-1
Menu System Integration	3.11-1
Return Codes	3.12-1

Chapter 4—System Tools

CSDISPLY — Computron’s Display Utility	4.1-1
Vutil – An Indexed File Utility	4.2-1
Program Report List	4.3-1
Display Computron Log File	4.4-1
Clear Print Pass File	4.5-1
Clear Hold List Entries	4.6-1

Chapter 5—Computron Server Administration

Computron Server Configuration Management - ADMTOOLS	5.1-1
Batch Processing Setup for Unix - csenq	5.2-1
Analyzer	5.3-1
Computron’s Configuration File	5.4-1
Creating Scripts via Computron’s Procedure Generator	5.5-1
Application Server Directory and File Structure	5.6-1
Computron’s Start Script	5.7-1
Computron’s Cron Job CSCRON	5.8-1

Chapter 6—Relational Database

RDBMS Utilities	6.1-1
RDBMS Loading and Unloading	6.2-1
Miscellaneous Utilities	6.3-1
Debugging Options	6.4-1
RDBMAP Format Maintenance	6.5-1
XMLMAP Format Maintenance	6.6-1
UEXMAP Format Maintenance	6.7-1
Server Log Files	6.8-1

Appendices

Appendix A: Printer Set Up	A-1
Appendix B: UserExits	B-1
Appendix C: LGMAP File	C-1
Appendix D: WISP Command Processor	D-1
Appendix E: Batch Processing Set-Up for Unix lp	E-1
Appendix F: Batch Processing Set-Up for Unix csenq	F-1
Appendix G: Configuration Change to Support a Terminal Server or Web Server	G-1

Preface

This guide provides information required for the operation and maintenance of Computron systems. The *System Administrator's Guide* is intended for use by the individual appointed as the system administrator for Computron applications. It is expected that this individual has experience with the Unix operating system. It does *not* assume extensive knowledge of Computron applications (e.g., General Ledger, Accounts Payable, etc.).

Prerequisites

The main prerequisite for using this manual effectively is that you have some hands-on familiarity with your Unix console, as well as a basic working knowledge of your Unix Server environment. In addition, the *User's Guide for WEBdesk* is highly recommended if you have no previous experience with Computron software or Computron's graphical WEBdesk presentation.

The *User's Guide for WEBdesk* for (for the Java based presentation or the Visual Basic® programming system presentation) documents the basic features shared by all Computron systems. For example, it outlines the basic types of windows that display in Computron functions and explains how to proceed from one window to the next. Other areas of discussion include:

- ◆ How to access dialog boxes, toolbar functions, menus, panels, etc.;
- ◆ The standard types of functions used to maintain system files and conduct inquiries;
- ◆ The Universal Line Method – a unique Computron feature, allowing for easy entry and manipulation of large volumes of data.

Organization of the Guide

The *System Administrator's Guide* provides information required for the day-to-day maintenance of Computron systems. This guide consists of six chapters. These chapters are primarily intended for use as reference material and consequently need not be read in any particular order. When you require information about a particular topic, turn directly to the section relating to that topic.

Chapter 1, Universal Utilities

This chapter provides detailed descriptions of the functions found on the standard Universal Utilities menu.

Chapter 2, Security Subsystem

This chapter provides an overview of the Computron security subsystem available in each application system. In addition, detailed descriptions of each function used to establish the operating parameters for the security subsystem are included.

Chapter 3, Job Processor

This chapter details the operation of the Job Processor module; used to proceduralize the running of Computron/non-Computron functions. For example, the Job Processor module is used to proceduralize report generation.

Chapter 4, System Tools

This chapter provides detailed explanations of various Computron system tools, including: Vutil – an indexed file utility; Program Report List; Clear Print Pass File and Clear Hold List Entries.

Chapter 5, Computron Service Administration

This chapter provides detailed explanations for administering and monitoring Computron services running under Unix, from within the Computron Server Admin Tool.

Chapter 6, Relational Database

This chapter provides detailed explanations of the Computron Database Utilities and of various troubleshooting steps.

The chapters are divided into sections, one section for each function. Each section is, in turn, subdivided into the following possible subsections (as appropriate for the material):

Introduction

This is a brief overview of the function, its purpose and operation.

Window Description

Contains detailed field descriptions for each function. The following information is supplied for modifiable fields:

- ◆ definition — how does what you enter affects the operation of the system
- ◆ possible field values — what to enter
- ◆ impact on other parameters
- ◆ other fields/parameters that have an impact on entries made in the field
- ◆ defaults — values displayed automatically by the system

- ◆ validation — the system checks.

In the *System Administrator's Guide*, fields that are “Q-Markable” are indicated by placing the Q-^{Mark} symbol beside the field name definition. The Q-^{Mark} facility gives you the ability to view and select from a list of valid codes for the field. Illustrations of the function windows are also included.

Additional Considerations

This refers you to other functions in the application being utilized and to other sources of information that relate to the specific task at hand. It also highlights special points that you need to consider while working with the function.

Report Description

This subsection appears only for those functions that generate reports as output. A sample of the report is included in this section and, where appropriate, a legend that can help you interpret the report contents is also included.

Sample Applications

Detailed sample applications are included for many functions, which can help to guide you in the development of your own applications of the function. Generally, this section describes some of the most common uses of the function.

Error Messages

Finally, this is a list of error messages, which are printed on output reports (where such reports are produced), is included. This subsection also contains a description of the conditions that would cause the error to occur, as well as corrective actions that you can take.

Depending upon the installation, several appendices may be included in the *System Administrator's Guide*. Appendices address subjects that apply to more than one function or serve as general background information.

Soft Technology

All Computron systems include the capability to tailor windows and reports without the need to develop customized software. This capability has been developed in order to tailor the system to a specific client needs. Soft technology includes the ability to:

- ◆ tailor prompts, error messages, and report formats
- ◆ change the order of fields within an entry window
- ◆ suppress fields
- ◆ set default values for fields
- ◆ maintain multiple versions of the same entry windows (e.g., one version in English and one version in French) within the same installation
- ◆ allow you to proceduralize functions
- ◆ restrict access to specific fields
- ◆ restrict access to operating modes.

In the AUI environment, the terminology and layout of fields or windows (or forms) may be tailored via the Visual Basic Forms Editor. Modifications to report formats as well as global terminology changes are accomplished via Soft Screen Maintenance.

All menus, windows and reports illustrated in this manual are considered “standard” for Computron’s systems. The field names used may not be the same as those appearing on your windows if your installation has incorporated the use of the Soft Technology utilities -Visual Basic Forms Editor and Soft Screen Maintenance.

Soft Technology utilities are not documented in the *System Administrator’s Guide*. The Visual Basic Forms Editor is documented in the *Forms Editor User’s Guide featuring the Visual Basic Programming System*; and Soft Screen Maintenance is documented in the *Soft Screens User’s Guide*.

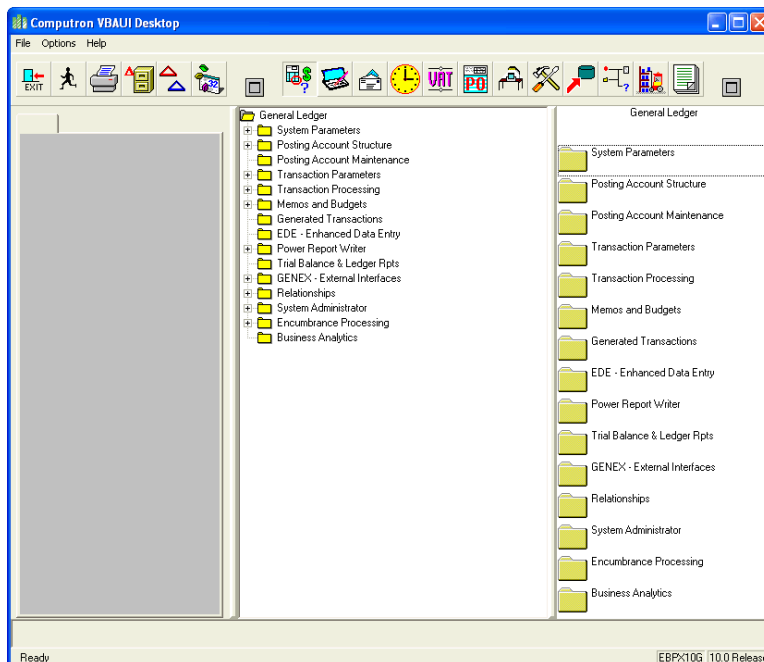
☞ *Note:* Soft Screen Maintenance may be included in the Universal Utilities menu for some installations.

Menus and Windows

System Administrator Standard Menus

The Computron Desktop is a powerful navigation tool that helps you to swiftly locate specific programs within application systems and then launch them with a click of your mouse. It also allows you to click-and-drag frequently used programs onto an *Alias* panel for quick launches.

The left portion of the *Navigator* panel shows groups and hierarchies of programs within menus (illustrated as file folders). Within each application system, individual programs (which perform specific accounting or support functions) are organized into their hierarchy of menus and submenus, as illustrated here:



General Ledger Main Menu

For example, most table maintenance functions are performed through programs—an online transaction entry is a program. Normally, similar programs (such as all transaction entry functions for an application

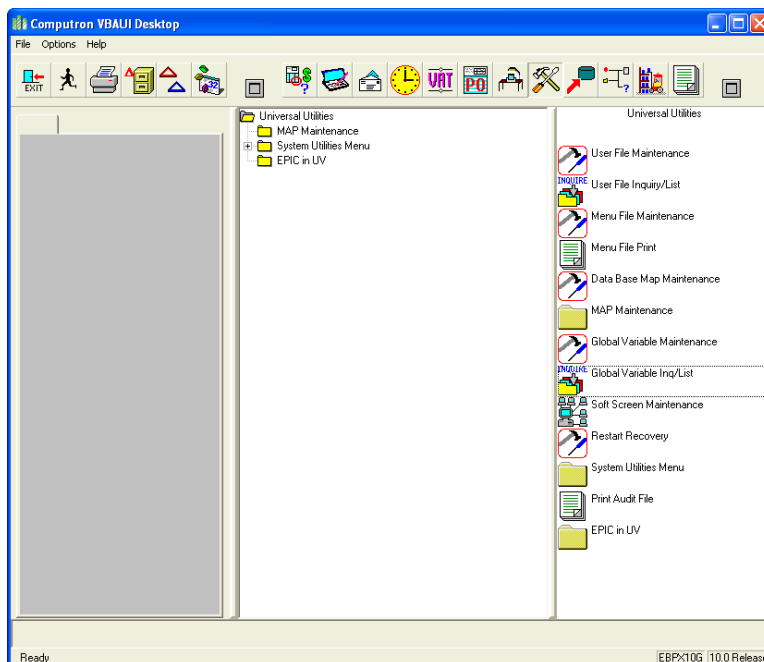
system) are grouped together. A list of available programs and their functions is available in the operation guide for each system.

There are several browsing methods you can use to locate the desired program:

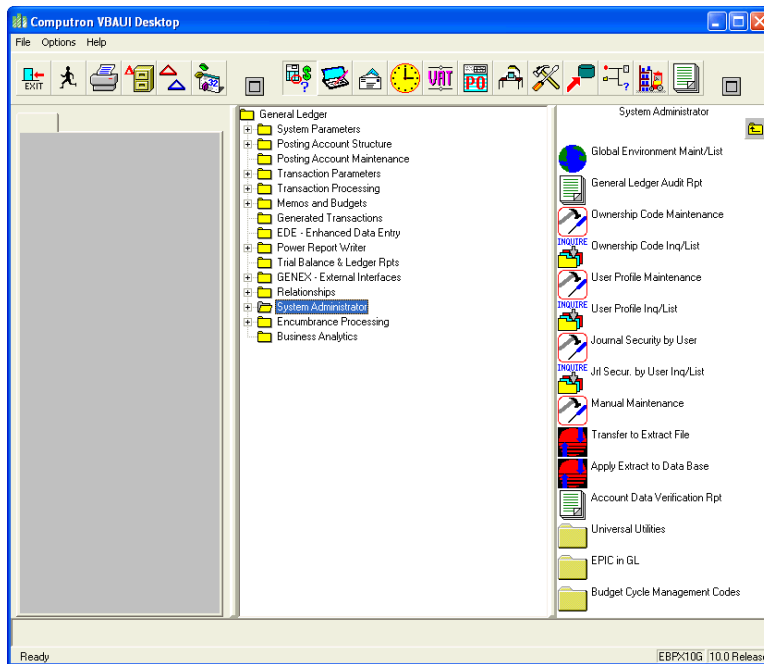
- ◆ Clicking a menu folder to display its contents in the Navigator list box.
- ◆ Clicking the plus symbol (+) next to a folder to display its submenus (if any) beneath; then clicking the submenu to display its contents in the Navigator list box. (Appears only with folders that contain submenus.)
- ◆ Clicking the minus symbol (–) next to a menu folder to hide its submenus; appears only when submenus are already displayed.

For detailed instructions on using the Computron Desktop, please refer to Chapter 2, “The Computron Desktop,” in the *User's Introduction Guide for AUI (featuring the Visual Basic programming system)*.

The Computron user is granted access to various application functions, depending upon the controls established for the organization. However, access to the following menus is normally granted only to high-level application users, the system administrator and the database administrator:



Universal Utilities Menu



General Ledger System Administrator Menu

The General Ledger System Administrator menu illustrates the types of functions allowed to you (the system administrator) for an application system. The types of functions provided to you within the application include purges, audit and productivity reports, maintenance of the global environment, as well as functions that support the Security Subsystem.

These Security Subsystems functions are documented in each application system's respective *Operations Guide* with the exception of Security Subsystem functions — Ownership Code and User Profile. Although these functions are defined at the application level, they are documented as a part of the overall Security Subsystem, see Chapter 2, "Security Subsystems," for additional information.

Window Types

In order to make our application systems as easy to use as possible, Computron features standardization of windows wherever possible; the window type is one of these standards. Each Computron application function window includes a type code, located in the lower right-hand corner of the window that specifies the window function.

Even though Soft Technology features may be implemented (which can make your windows appear different from the illustrations in this manual), the window type code that appears in the function window always corresponds to the one in the illustration.

The following illustrations of each different window type include descriptions of the type of processing performed by each.

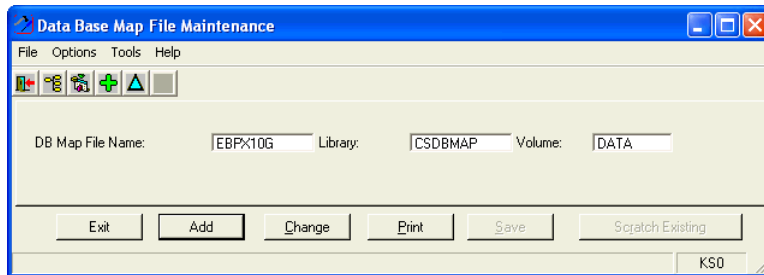
Key Windows (KS)

A key is a unique identifier for a data record (e.g., a company number, account number, currency code, etc.). Key windows are used to enter key information that the system uses to locate data records and, on

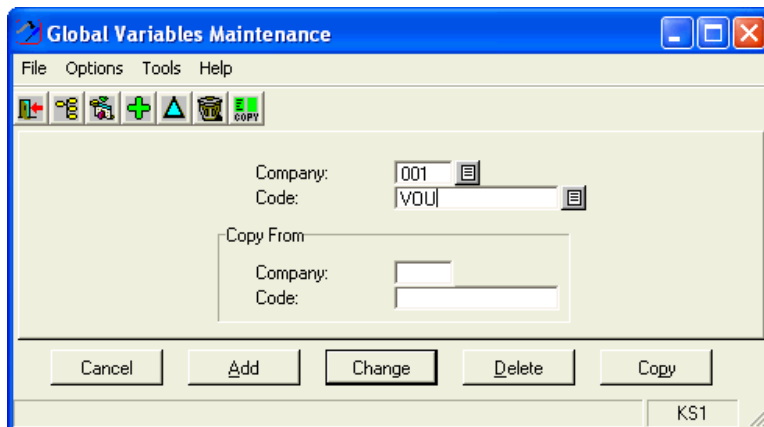
some Key windows, to select processing operations (e.g., add, change, delete, copy). Anytime you want to enter or change information for that data record, you must enter the key information on a Key window.

Key windows are designated by a window ID that starts with *KS*. They typically include the following areas and tools:

- ◆ summary area
- ◆ data entry area
- ◆ command buttons (to perform functions)



KS0 Window - Key Window

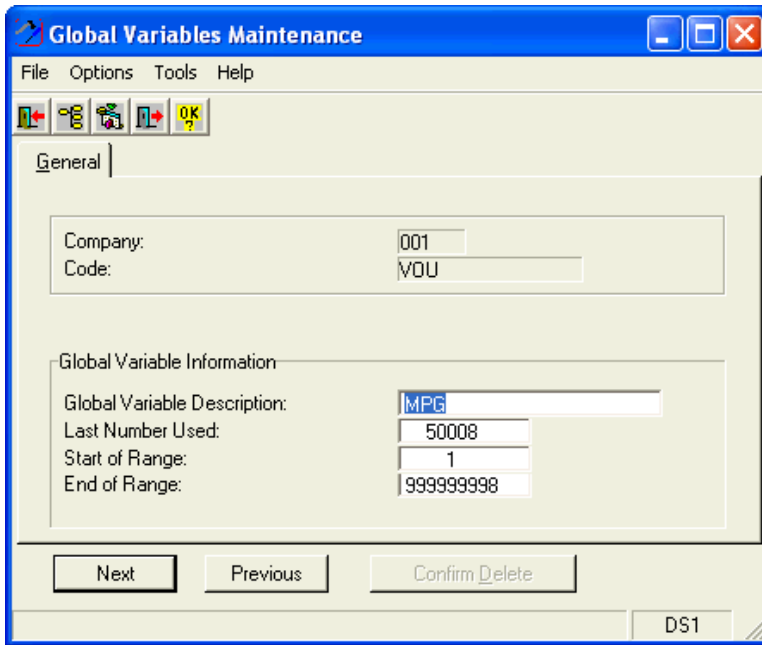


KS1 Window - Key Window

Data Windows (DS)

Data windows are used to enter data into the system and are typically associated with maintenance functions. Generally, this type of window is used for functions through which a small volume of data is updated.

Normally, when multiple DS windows are required to complete a given entry, the windows are numbered sequentially for identification purposes (e.g., DS1, DS2, etc.).

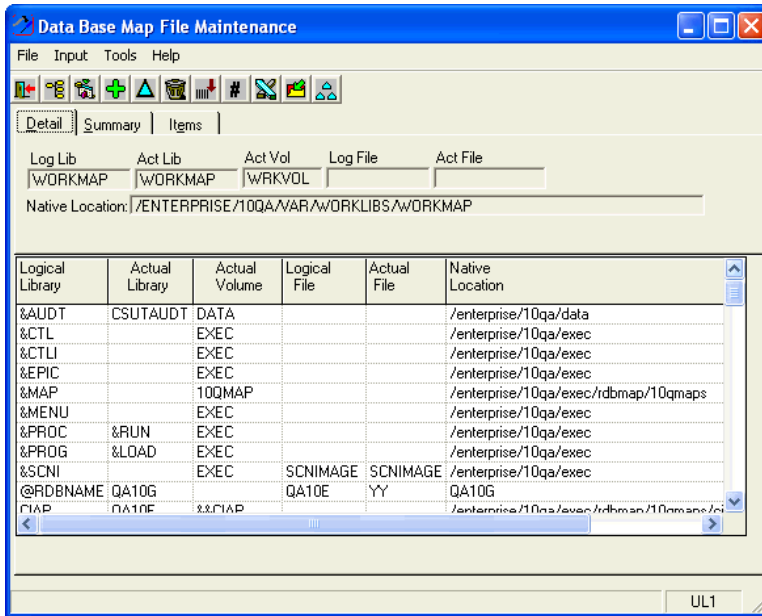


DS1 Window - Data Window

☞ *Note:* DS windows are sometimes displayed in inquiry functions to illustrate the window through which the data was originally entered.

Universal Line Method Windows (UL)

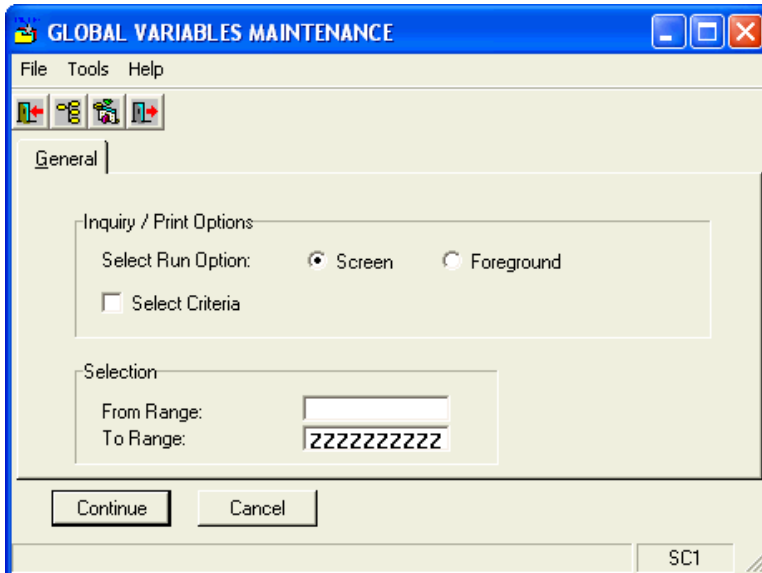
Universal Line Method windows are used to enter large volumes of data for functions like Data Base Map File Maintenance. This window allows you to enter multiple lines of data on a single window, speeding up the data entry process. Usually, this type of window is used to enter data that is repetitive in nature (e.g., Data Base Map records) and includes a small number of prompts (one or two lines).



UL1 Window - Universal Line Method Window

Selection Criteria Windows (SC)

Selection Criteria windows are used to select data for a report/inquiry or updating process. You can enter a range of user ID's, account numbers, companies, journals, etc., selecting them for additional processing. For example, when you are generating a listing of posting accounts in the database, you may specify to list only posting accounts for a range of departments.

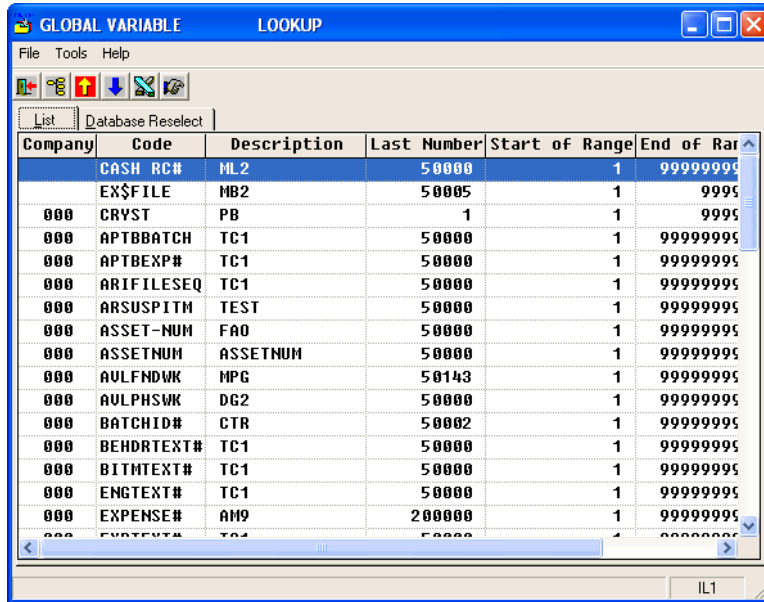


SC1 Window - Selection Criteria Window

Clicking Continue (or pressing Enter) either displays any additional selection windows (SC99) for the function or immediately starts the selection process (e.g., report, inquiry, or update).

Inquiry List Windows (IL)

Inquiry/List windows display data lists. For some functions, IL windows are display-only windows (allowing you to view data without performing any processing). For other functions, you may be able to select individual items on the windows for additional processing or inquiry.



Company	Code	Description	Last Number	Start of Range	End of Range
	CASH RC#	HL2	50000	1	99999999
	EX\$FILE	MB2	50005	1	9999
000	CRYST	PB	1	1	9999
000	APTBBATCH	TC1	50000	1	99999999
000	APTBEXP#	TC1	50000	1	99999999
000	ARIFILESEQ	TC1	50000	1	99999999
000	ARSUSPITM	TEST	50000	1	99999999
000	ASSET-NUM	FA0	50000	1	99999999
000	ASSETNUM	ASSETNUM	50000	1	99999999
000	AULFNDWK	MPG	50143	1	99999999
000	AULPHSWK	DG2	50000	1	99999999
000	BATCHID#	CTR	50002	1	99999999
000	BEHDRTEXT#	TC1	50000	1	99999999
000	BITHTEXT#	TC1	50000	1	99999999
000	ENGTEXT#	TC1	50000	1	99999999
000	EXPENSE#	AM9	200000	1	99999999
000	EXPTEXT#	TC1	50000	1	99999999

IL1 Window - Inquiry/List Window

Chapter 1

Universal Utilities

1.1 Overview: Setting Up Computron Users

Computron requires Unix user accounts that create Computron processes to belong to a unique user group (usually *ctron*). This section provides a high-level overview of the steps necessary to establish user accounts for Computron software running in a Unix environment while using a relational database management system (RDBMS). These instructions include the Unix command-line syntax used for the generic SUN and HP-UX systems, as well as the special syntax required for the AIX system.

☞ *Notes:*

- ◆ AIX system administrators can also use the *SMIT* GUI interface to create Computron user accounts.
- ◆ HP-UX system administrators can also use the *SAM* GUI interface to create Computron user accounts.

Creating Unix User Accounts

Before installing Computron software onto the Unix server that will serve as your Computron application server, you should follow these setup procedures:

1. A Computron group (usually *ctron*).
2. A Computron administrative user ID (usually *ctronsys*).
3. User accounts for all Computron application users in the Computron group.
4. Establish user account passwords.
5. Copy the user *.profile* files to the users' home directories.

☞ *Important!* Before adding Computron user accounts, please note the following caveats:

- ◆ The Unix administrator *must* have root permissions when creating the Computron Unix groups and accounts.
- ◆ Check with your local system administrator(s) to make sure that any established user/group ID conventions and standards are followed. For example, it may be preferable for Computron user/group IDs to be set up in a specific range of values.
- ◆ If NIS (Network Information Service) is being used to manage the password file across the network, only personnel who are familiar with the established security conventions should be allowed to add the necessary Computron user and group IDs.

Step 1: Create the Computron Group

First, you should establish a Computron group (usually *ctron*).

Generic Unix:

1. On the command line, use the following syntax:

```
groupadd [-g <gid> [-o]] <group>
```

Example: a typical command line would appear as follows:

```
groupadd -g 202 ctron
```

2. This command creates a `ctron::202:` entry in the `/etc/group` directory.

Special AIX:

1. On the command line, use the following syntax:

```
mkgroup [-a] [-A] "attr=value" ... newgroup
```

☞ *Note:* For *attr* command explanations, refer to the online Unix *man* pages.

Example: a typical command line would appear as follows:

```
mkgroup id=202 ctron
```

2. This command creates a `ctron!:202:` entry in the `/etc/group` directory.

Step 2: Add the Computron Administrator Account

Next, add the Computron administrative account (usually *ctronsys*) to the *ctron* group.

Generic Unix:

On the command line, use the following syntax:

```
useradd [-u <uid>] [-o]] [-g <group>] [-G <group> [<group...>]]  
[-d <dirs>]
```

Example: a typical command line would appear as follows:

```
useradd -u 203 -g 202 -d /export/home/ctronsys -s /bin/ksh -m -  
c "Computron Administrator" ctronsys
```

Special AIX:

On the command line, use the following syntax:

```
mkuser [-a] "attr=value" ... newuser
```

Example: a typical command line would appear as follows:

```
mkuser id=203 pgrp=ctron home=/home/ctronsys shell=/bin/ksh  
gecos='Computron Administrator' ctronsys
```

☞ *Note:* For *attr* command explanations, refer to the online Unix *man* pages.

Step 3: Add the Computron User IDs

Next, add the Computron application user accounts to the *ctron* group.

Generic Unix:

On the command line, use the following syntax:

```
useradd [-u <uid>] [-o]] [-g <group>] [-G <group>  
[<group...>]] [-d <dirs>]
```

Example: a typical command line would appear as follows:

```
useradd -u 203 -g 202 -d /export/home/jrd -s /bin/ksh -m  
-c "John Doe" jrd
```

AIX:

On the command line, use the following syntax:

```
mkuser [-a] "attr=value" ... newuser
```

Example: a typical command line would appear as follows:

```
mkuser id=203 pgrp=ctron home=/home/jrd shell=/bin/ksh  
gecos='John Doe' jrd
```

☞ *Note:* For *attr* command explanations, refer to the online Unix *man* pages.

Step 4: Establish Account Passwords

Generic Unix:

After creating each user account, assign the account password, as follows:

```
passwd jrd
```

You are then prompted to enter a password for the account. We recommend assigning the user a generic password and then allowing them to change it when they log in initially.

AIX:

After creating each user account, assign the account password, as follows:

```
passwd jrd
```

You are then prompted to enter a password for the account. On AIX, when *root* changes the user password, the password is pre-expired, which forces the user to change the password at the next login.

Step 5: Copy User .profile Files

To assist System Administrators when setting up Unix user accounts for Computron users, Computron supplies a sample .profile file (*/.../profiles/users* [where */...* represents the directory name of the file system where the Computron software resides]) for general Computron users. When copied to the **.profile** file, in the user's home directory, this file initializes the Computron environment for the user upon login and automatically starts the application software. Additionally, Computron supplies a sample **.profile** file (*/.../profiles/ctronsys*) for the Computron Administrator, **ctronsys**. When this file is copied to the home directory of the user **ctronsys**, the Computron environment is initialized and the user **ctronsys** is placed at a Unix shell prompt upon log in.

Add Users to the RDBMS

For Unix/RDBMS implementations (Oracle, Sybase, and Informix), the database administrator should now add user IDs and passwords for the Computron users that will access the database, including the Computron system administrator ID (*ctronsys*).

Once your users have been added, you can assign database permissions, making sure to grant table owner privileges to the *ctronsys* DB ID.

☞ *Note:* Please refer to your RDBMS documentation for instructions on establishing user IDs and assigning database permissions. For detailed instructions on establishing a relational database environment for Computron software, refer to Computron's *Database Configuration Instructions* for your RDBMS.

Add Users to the Computron User File

Before adding users with the Computron system, you should make sure that Computron's Client/Server software has been installed and correctly configured on the Unix server.

Install the Computron Software

For detailed instructions of installing Computron software, refer to the *Computron Installation Instructions for UNIX/RDBMS*.

Initialize the Server for WEBdesk Client

For detailed instructions of installing Computron software, refer to the *Computron Installation Instructions for WEBdesk*.

User File Maintenance

Computron user IDs are added to the Computron User file (USERFIL3) via the User File Maintenance function, which is located on the System Utilities menu. Computron's User file contains one record for each user permitted access to the Computron applications.

The User file is designed to complement the Unix Password file, which is part of the Unix system security. The Unix Password file indicates the valid user accounts that can log onto the Unix system. Computron's User file, in turn, indicates which of those individuals in the Unix Password file can access Computron functions. Because the files are related in this manner — the User file maps the Unix user account name to Computron's user ID

Detailed instructions on establishing and maintaining users in the Computron User file are located in Section 1.2, "User File Maintenance," later in this chapter

1.2 User File Maintenance

Introduction to User File Maintenance

The User File Maintenance function allows you to establish and maintain records in the User file. Computron's User file contains one record for each user permitted access to the Computron applications. The User file is designed to complement the Unix Password file, which is part of Unix system security. The Unix Password file indicates who can log onto the Unix system. Computron's User file, in turn, indicates which of those individuals in the Unix Password file can access Computron functions. Because the files are related in this manner — the User file maps the Unix user account name to Computron's user ID; Computron recommends that the first three characters of a Unix user account name be unique, since the Computron user ID is dependent upon Unix system security.

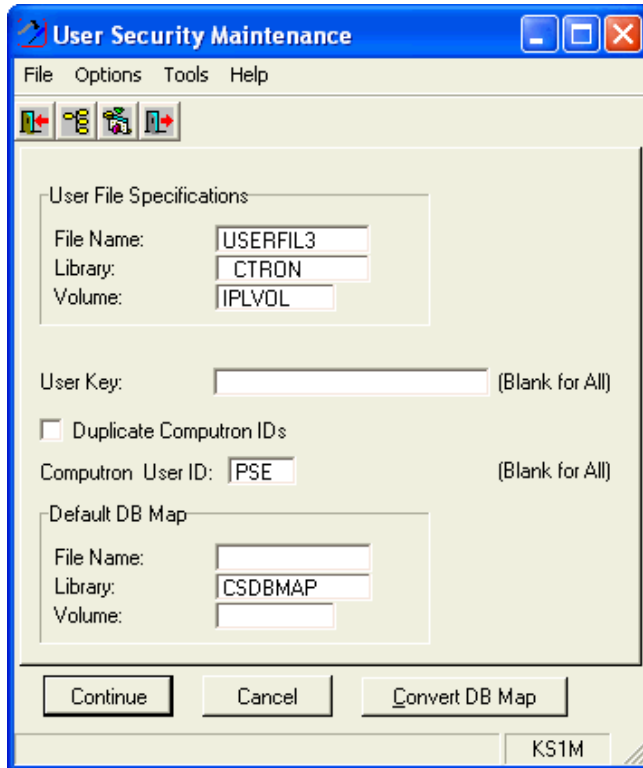
☞ *Note:* If a user without a User file record attempts to log on to Computron, the following message displays: "Invalid User ID or Password. Please try again."

The User file serves as the source of authentication for both Computron applications and the relational database accessed via the Computron applications. Additionally, Computron's User file contains security parameters for each Computron user. These parameters, used in conjunction with other components of the security subsystem (menus, security profiles and screen images), establish user access privileges throughout the Computron applications.

● *Important!* Relational database installations require a special Computron user record, &&A, for every owner of Computron tables. This record is used to connect to the database when either creating Computron tables or their stored procedures. Without it, various errors can occur within the Computron applications. For detailed instructions on adding the &&A user record, see "Special Database User ID Records," at the end of this section.

User Security Maintenance Window (KS1)

The User Security Maintenance Window (KS1) is the first window to display in User File Maintenance. This window prompts for default information to assign to User file records.



User Security Maintenance Window (KS1)

File Name

This is the name of Computron's User file, USERFIL3. This field is non-modifiable.

Library

This is the directory where the User file resides.

Default: _CTRON_

The file, USERFIL3, resides in the Unix directory, `\...\exec_ctron_` (where `\...` represents the directory name where the Computron software resides); thus, the library, `_CTRON_`, represents the directory `_ctron_`.

Volume

This is the six character logical name that defines the directory where the selected library resides.

Default: IPLVOL

Since the file, USERFIL3, resides in the Unix directory, `\...\exec_ctron_` (where `\...` represents the directory name where the Computron software resides), and the library, `_CTRON_`, represents the directory `_ctron_`, any volume defined in Computron's LGMAP as the path `\...\exec` can be entered as the volume.

☞ *Note:* Computron releases an LGMAP file (in the `\...\exec_ctron_wisp\config` directory on the server) that contains the following entry:

```
IPLVOL\...\exec\_ctron\_userfil3
```

See Appendix C, “LGMAP File,” for additional information.

User Key

This is the Unix user account name of the record to maintain. Enter a specific user key to maintain the corresponding record or leave the field blank to maintain all of the records in the User file (i.e., to maintain multiple users).

Duplicate Computron IDs

This indicates whether multiple records can be generated in the User file for the same Computron ID. For example, all data entry clerks for the Accounts Payable department can be established with the same Computron ID. Valid options are:

- Duplicate Computron ID's are allowed.
- Duplicate Computron ID's are not allowed.

Default:

☞ *Note for RDBMS Installations:* This check box must be selected when adding the special &&A user record for every owner of Computron tables. For detailed instructions on adding the &&A user record, see “Special Database User ID Records,” at the end of this section.

Computron User ID

This is the Computron user ID of the record to maintain. Enter a Computron user ID to maintain a specific record or leave the field blank to maintain all of the records in the User file, i.e., to maintain multiple users. The User file record can be maintained by the Computron user ID or the user key.

Default: The current user's Computron ID.

The following Default DB Map fields are commonly used when adding a large number of records to the User file. The values in these fields supply defaults for the subsequent group of IDs added via the User Logon Security Maintenance Window (UL1).

File Name

This is the name of a Database Map file that serves as the default Database Map file for any records subsequently added to the User file. The database map identifies the location of data for the Computron applications.

Library

This is the directory that contains the current database map. This directory serves as the default Database Map library for any records subsequently added to the User file. Computron conventionally stores all Database Map files in the directory — CSDBMAP.

Volume

This is a six-character logical name that defines the location of the directory where the database map resides. This volume serves as the default Database Map volume for any records subsequently added to the User file.

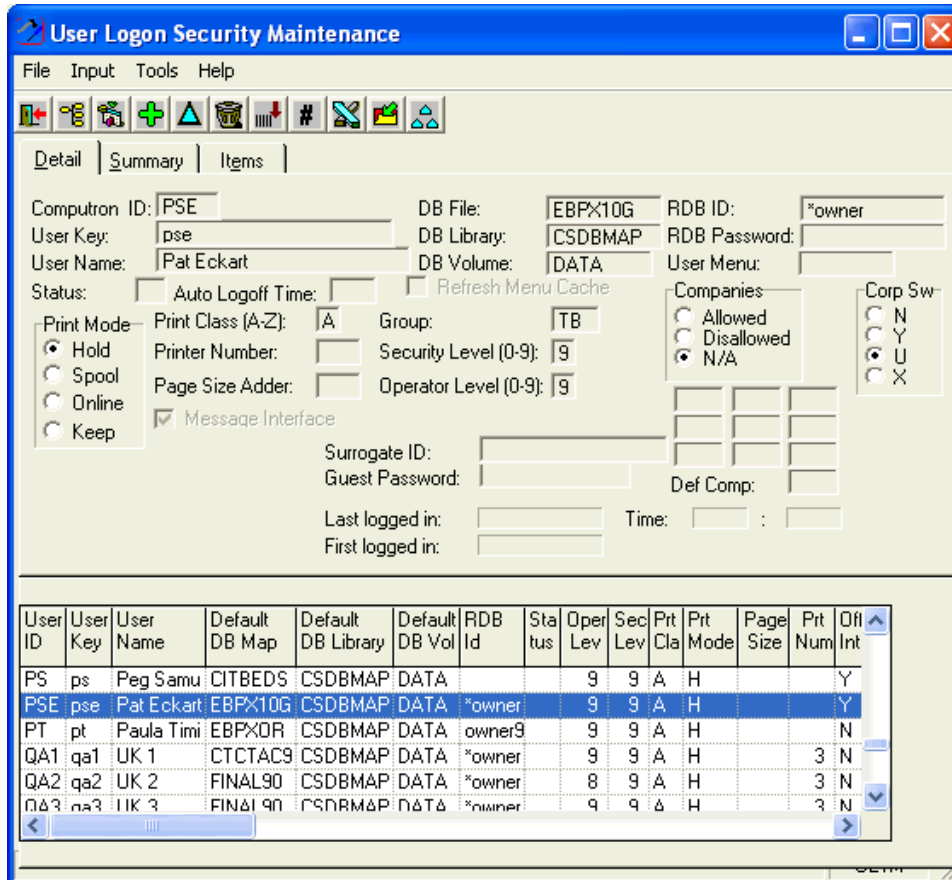
Note: Normally in relational database implementations, Computron systems are installed with the database map on the same volume as any parameter data files, so that it is not necessary to specify the volume in the database map for files/libraries on the same volume as the Database Map file

Table 1.1-1	
User Security Maintenance Window (KS1) Options	
Button	Description
Continue	Proceed to the User Logon Security Maintenance Window (UL1) where User file records are created, modified, and deleted.
Cancel	Exit User File Maintenance.
Convert DB Map	Proceed to the User Security Maintenance Window (KS2) where one or more of the components of the database map location (e.g., Name\Library\Volume) can be changed for a specific record or for all records in the User file. <i>Note:</i> To process all records in the User file, leave the User ID field on the User Security Maintenance Window (KS1) blank.

Enter the appropriate information on the User Security Maintenance Window (KS1) and then click Continue to display the User Logon Security Maintenance Window (UL1).

User Logon Security Maintenance Window (UL1)

The User Logon Security Maintenance Window (UL1) allows you to process specific records in the User file on a user-by-user basis.



The screenshot shows the 'User Logon Security Maintenance' window with the following fields and options:

- Computron ID: PSE
- User Key: pse
- User Name: Pat Eckart
- Status: Auto Logoff Time: Refresh Menu Cache
- DB File: EBFX10G
- DB Library: CSDBMAP
- DB Volume: DATA
- RDB ID: *owner
- RDB Password:
- User Menu:
- Print Mode: Hold, Spool, Online, Keep
- Print Class (A-Z): A
- Printer Number:
- Page Size Adder:
- Group: TB
- Security Level (0-9): 9
- Operator Level (0-9): 9
- Message Interface:
- Companies: Allowed, Disallowed, N/A
- Corp Sw: N, Y, U, X
- Surrogate ID:
- Guest Password:
- Def Comp:
- Last logged in: Time: :
- First logged in:

User ID	User Key	User Name	Default DB Map	Default DB Library	Default DB Vol	RDB Id	Sta tus	Oper Lev	Sec Lev	Pri Cla	Pri Mode	Page Size	Pri Num	Ofi Int
PS	ps	Peg Samu	CITBEDS	CSDBMAP	DATA			9	9	A	H			Y
PSE	pse	Pat Eckart	EBFX10G	CSDBMAP	DATA	*owner		9	9	A	H			Y
PT	pt	Paula Timi	EBFXDR	CSDBMAP	DATA	owner9		9	9	A	H			N
QA1	qa1	UK 1	CTCTAC9	CSDBMAP	DATA	*owner		9	9	A	H		3	N
QA2	qa2	UK 2	FINAL90	CSDBMAP	DATA	*owner		8	9	A	H		3	N
QA3	qa3	UK 3	FINAL90	CSDBMAP	DATA	*owner		9	9	A	H		3	N

User Logon Security Maintenance Window (UL1)

Computron ID

The Computron ID of the user. This field can be modified only when adding a record.


User Key

The Unix user account name of the user. This field can be modified only when adding a record.

Important! Relational database installations require a special Computron user record, &&A, for every owner of Computron tables. For example, the *ctronsys* ID would have a record named, &&ACTR. This record is used to connect to the database when creating Computron tables or their stored procedures. Without it, various errors may occur within the Computron applications. For detailed instructions on establishing the &&A user record, refer to Computron's *Database Administrator's Guide*.

User Name

This is the name of the user. This name, which can be up to twenty-four alphanumeric characters, is displayed in the top left-hand corner of the menu screens used by Computron's character cell presentation and on the desktop presentation.

 *Note:* Computron recommends that you do not use special characters in this name except where indicated for special users. Single quotes, double quotes, etc. can create errors when running scripts.

Status

To help you keep track of the entries made on this window, this field indicates the status of the user record during this session. This non-modifiable field is automatically updated by the system. Valid options are:

A – Added.

C – Changed.

D – Deleted.

When you click the Process/Save button, the indicated status takes effect.

Example: A user record added during a session displays the letter A in this field.

Auto Logoff Time

This field sets the default value for WEBdesk Preference's Auto Logoff Timeout in Minutes field. The WEBdesk auto-logoff requires you to re-authenticate your access to the system but does not kill any currently running processes.

Refresh Menu Cache

This is a security feature that indicates whether a user's local menu cache is automatically updated by the server upon logging onto the Desktop. This option overrides the Get New Menus Direct preference setting in the ctc.ini file on the PC.

Do not update the local menu cache upon logging onto the Desktop.

Always update the local menu cache upon logging onto the Desktop.

Default:

DB File

This is the name of the user's default Database Map file. The database map identifies the location of application data. If adding a record, the entry in the default DB Map File Name field on the previous window appears as a modifiable default in this field.

DB Library

This is the directory that contains the current database map. Computron conventionally stores all Database Map files in the directory — CSDBMAP. If adding a record, the entry in the Default DB Map Library field on the previous window appears as a modifiable default in this field. If left blank, the system assumes the input values from usage constants at run time.

DB Volume

This is a six-character logical name that defines the location of the directory where the database map resides. If adding a record, the entry in the Default Database Map Volume field on the previous window appears as a modifiable default in this field.

Notes:

Normally, Computron systems are installed with the database map on the same volume as the parameter data files.

The database map location entered here is displayed at the top of the menu screens used by Computron's character cell and XML Web presentations and at the bottom of the menu windows in VBAUI and Java Web presentations.

The default database map location is only a default. Depending on the value of the Corp field (later), the user may be granted access to the change database map option.

The following DB ID and Password fields are used to log the Computron user onto the relational database. Depending on the values entered in these fields, the user is either logged directly into the relational database or is prompted to enter required authentication information.

DB ID

The relational database logon ID for the Computron user.

Note: An interactive value — *PROMPT can be entered in this field. An interactive value causes the system to prompt the Computron user for information required to connect to the relational database; upon initiating a Computron function that requires access to the relational database. The value *PROMPT, causes the system to prompt the user to enter an ID to log onto the relational database.

Note: The value — *OWNER placed in the DB-ID field forces the system to read the &&A record in the User file in order to determine the DB-ID and Password. By using this feature, when it is necessary to change the password for the database schema it only needs to be changed in one place, not in every individual user record. The &&A record contains the name of the relational database table owner in the Database ID field and the owner's database password in the DB Password field.

Password

This field is used in conjunction with the DB ID field to log the user onto the relational database. Enter either the password that corresponds to the user's DB ID or an interactive value, as described here. Once the user record is saved, the password is encrypted for storage. Interactive values cause the system to prompt the user for information required to log onto the relational database when initiating a Computron function that requires access to the relational database. Valid interactive entries are:

*PROMPT – This value causes the system to prompt the user for the appropriate relational database password.

*NOID – This value causes the system to prompt the user to enter the relational database password that corresponds to the DB ID entered in the Computron User file; the user's DB ID, however, is not displayed on the window.

*SHOWID – This value causes the system to display the user's DB ID as entered in the Computron User file and to prompt the user to enter the corresponding relational database password.

User Menu

This is the name of the menu file to display for this user upon logging onto the Computron application software. Any menu can be displayed to a user upon logging on, provided the menu file resides in the library, CIUVMENU. Specifying an initial menu for a user enables you to restrict the user's access to only those functions that are available from the initial menu. When left blank, the initial menu displayed to the user upon logging on is the menu file START.

For more information about printing, refer to Appendix A, "Printer Set Up."

Print Mode

This field determines the mode of printing.

Print Class

This field determines the print class.

Printer Number

This field determines the printer number.

Page Size Adder

This field is the number of additional lines added to a report to change the paper format, e.g., A4 size paper.

Message Interface

This field determines whether the user can send messages via the Unix system.

Group

This field is used to assign users a group code. When groups are used, group level security can be implemented for menus and windows where Computron's AUI Forms Editor (Visual Basic) is in use. This type of security enables you to allow/disallow group access to menu options and windows.

Security Level

This is the user's security level. The user's assigned security level controls access to Computron applications at the menu level.

This field, which can contain a number from 0 to 9, determines whether a user can access a menu selection. If the security level of a menu selection is greater than the security level of a particular user, the function is inaccessible to that user. In fact, it does not even display as an option on that user's menu.

Note: Security levels for menu selections are assigned via Computron's Menu File Maintenance function. See Section 1.4, "Menu File Maintenance," for additional information.

Example: Consider the Global Environment Maintenance function in Computron's General Ledger system. This function allows the user to establish corporate-wide parameters that direct all processing within the system. Because of the supervisory nature of this function, only a few individuals are to be allowed to access it. Suppose the function had been assigned a Security Level of nine in Menu File Maintenance. To make it inaccessible to most General Ledger users, assign the user a security level lower than nine.

Operator Level

This is the user's operator level. Operator level is a security feature designed to be used in conjunction with Computron functions. This field, which can contain a number from 0 to 9, indicates the level of access that the user has to specific buttons and individual fields on windows. The user's operator level is compared to the operator level of the buttons and data entry fields at run time. A button or data entry field is inaccessible to the user if its operator level is greater than the user's operator level.

Note: Operator levels are assigned to buttons and data entry fields on windows via AUI Forms Editor (Visual Basic).

Example: Suppose you want to restrict a user from deleting records in the Vendor Maintenance function of the Accounts Payable system. Assign the Delete button an operator level value higher than operator level of the user.

Companies

This field is used in conjunction with the company array fields (see later), this field indicates whether the user is allowed/disallowed access to the company list. This field is used if either the N or X option has been selected for the Corp field. Valid options are:

Allowed – The user is allowed to access only those companies entered in the company array fields.

Disallowed – The user is not allowed to access those companies entered in the company array fields.

N/A – The user is allowed access to all companies.

Default: N/A

unlabeled 9-field company array

These fields are used in conjunction with the Companies field. The companies listed in this array are either allowed or disallowed for this user. Up to nine companies can be specified.

Def Comp

This is the company number to be used as the user's default value for all prompts requesting a company number in Computron applications. When the default value is displayed, the user can either accept the default as established in this field or override it by entering a new company number or selecting the Change Company option.

Note: The user can override this default value. The User Default Change Window allows the user to enter a new value in the Default Company Number field. This new value is effective until the user logs off the system or changes it again. The advantage of the default is that it speeds up data entry.

Example: Suppose company 300 is entered in this field. When the user accesses the Voucher Entry function, the system begins processing for company 300.

Corp Sw

This field offers two additional levels of security. A user can be restricted from:

- ◆ Printing corporate reports; that is, reports spanning multiple companies as defined in the user's (Computron) financial system. Therefore, the user can run reports only for

individual companies, rather than all companies (including corporate reports). This option is commonly coupled with the Companies field (see earlier).

- ◆ Accessing the Change Database Map window to switch Database Map files. The Change Database Map window is accessed by selecting the toolbar button.

Valid options are:

Option	Corp Reports Allowed	Allow Access to DB Map
N	No	No
Y	Yes	No
U	Yes	Yes
X	No	Yes

Surrogate ID Guest Password

These fields represent credentials which identify the operating system account under which Financials processes are run. A given user may log in as a Guest, which means that he or she has his/her user context shifted to the ID defined in the Surrogate ID and Guest Password fields.

The Surrogate ID and Guest Password fields are also used as part of the Computron LDAP authentication mechanism.

Choose a maintenance option (Add, Change, Delete, Insert, Change Many, Process/Save and Exit) as appropriate.








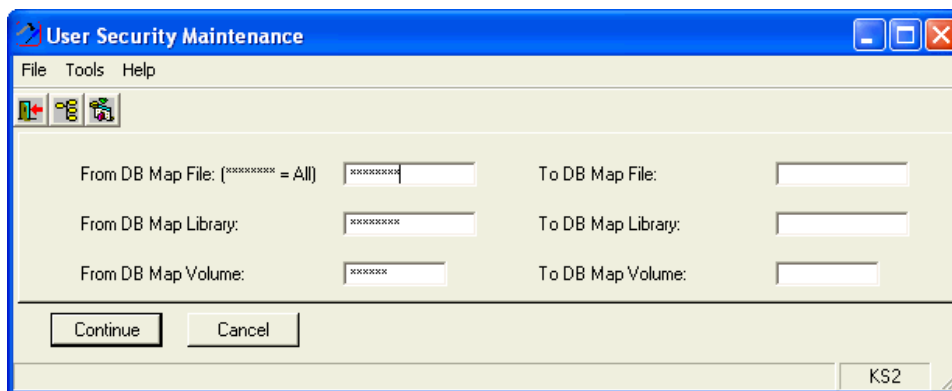
Table 1.1-2 Toolbar Options — User Logon Security Maintenance Window (DS1)		
Button	Function	Description
	Exit	Return to the User Security Maintenance Window (KS1) without saving any of the entries made.
	Add	Add a user record. This option presents the user currently highlighted in the list as a starting point for the new user. Once the new user's fields are updated, click the Save icon.
	Change	Change a user record. Highlight the line to change and then click the Change button. Once the changes are complete, click the Save icon.
	Delete	Delete a user record. Highlight the record to delete and click the Delete button. Once the delete is confirmed, click the Save icon.

Table 1.1-2 Toolbar Options — User Logon Security Maintenance Window (DS1)		
Button	Function	Description
	Insert	Insert a user record. Highlight the record above which you wish to insert the new user record and click the Insert button. <i>Note:</i> Users are displayed alphabetically by User Name.
	Process/Save	Save the User file with all of the entries you made during this session.
	Change Many	Change many user records. Highlight the first record to change and click the Change Many button. When the first record is completed, the next record is displayed in a modifiable format, allowing you to make changes, etc. Once the changes are complete, click the Save icon.

User Security Maintenance Window (KS2)

The User Security Maintenance Window (KS2) displays when you click the Convert DB Map button. This window allows you to change the Database Map file location for all of the records in the User file. This is a particularly useful feature because it allows the user to make global changes to the User file, without having to access individual records.

Example: Suppose all Computron users initially point to a test database. When the system goes live, you can automatically switch all users to the live database at this window.



User Security Maintenance Window (KS2)

The database map location is identified by three components — File, Library and Volume. The conversion feature allows you to modify any one of these components (e.g., File, Library or Volume) as necessary to identify the new Database Map file.

For each component of the database map location, this window prompts you for the original (from) value, and new (to) value to replace the original. Simply enter values in the appropriate *From* and *To* fields and

click the Continue button. The system proceeds to the Universal Line Method Window (UL1) where you can review the changes.

⚠ *Warning! Because all matching records in the User file are changed, you should double-check the existing user records before making a group change.*

Example: Suppose you want to change the database map volume from TSTVOL to DATVOL. After entering this change on the conversion window (KS2) and processing the Universal Line Method Window (UL1), the User file is updated for all records. All of the records with a volume of TSTVOL are assigned a volume of DATVOL, regardless of their Database Map file and library entries.

The safest way to use this feature is to identify the From DB Map, Library and Volume (e.g., WHMAPSV, CIGLDEMO, TSTVOL) and the To DB Map, Library and Volume (WHMAPSV, CIGLLIVE, DATVOL). Thus, only those pointers to CIGLDEMO are updated to CIGLLIVE.

☞ *Note:* To avoid any data corruption, ensure no users are processing using the From DB Map file at the time of the change.

See Section 1.6, “Database Map Maintenance,” for additional information.

Special Database User ID Records

This section discusses the special User file requirements for RDBMS implementations.

The &&A User Record

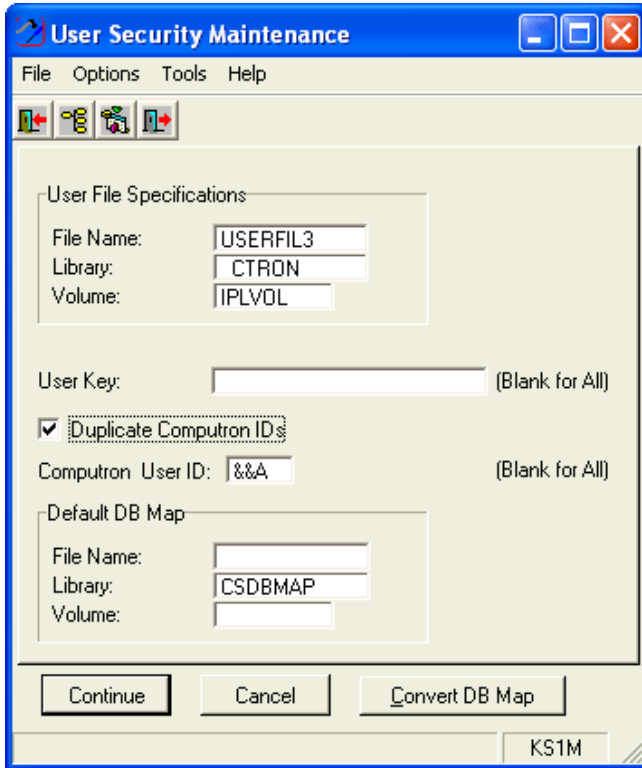
All RDBMS installations require a unique Computron user record, &&A, for each user that is an owner of Computron tables. Commonly referred to as *amper-amper-A*, the &&A record is used to connect to the database when either creating Computron tables or their stored procedures. Without it, various errors may occur within the Computron applications.

Before you begin, please note the following guidelines:

- ◆ The Allow Duplicates check box on the User Security Window (KS1) must be selected in order to create an &&A record for each user that is an owner of Computron tables.
- ◆ You cannot create duplicate &&A records for the same table owners (e.g., tables owners with the same ID on two different database servers). This is because the User File on the Computron application server always uses the first &&A record that it finds.
- ◆ Table owners must have a security level of 9 in order to run Computron’s database utilities (e.g., csunload_data).

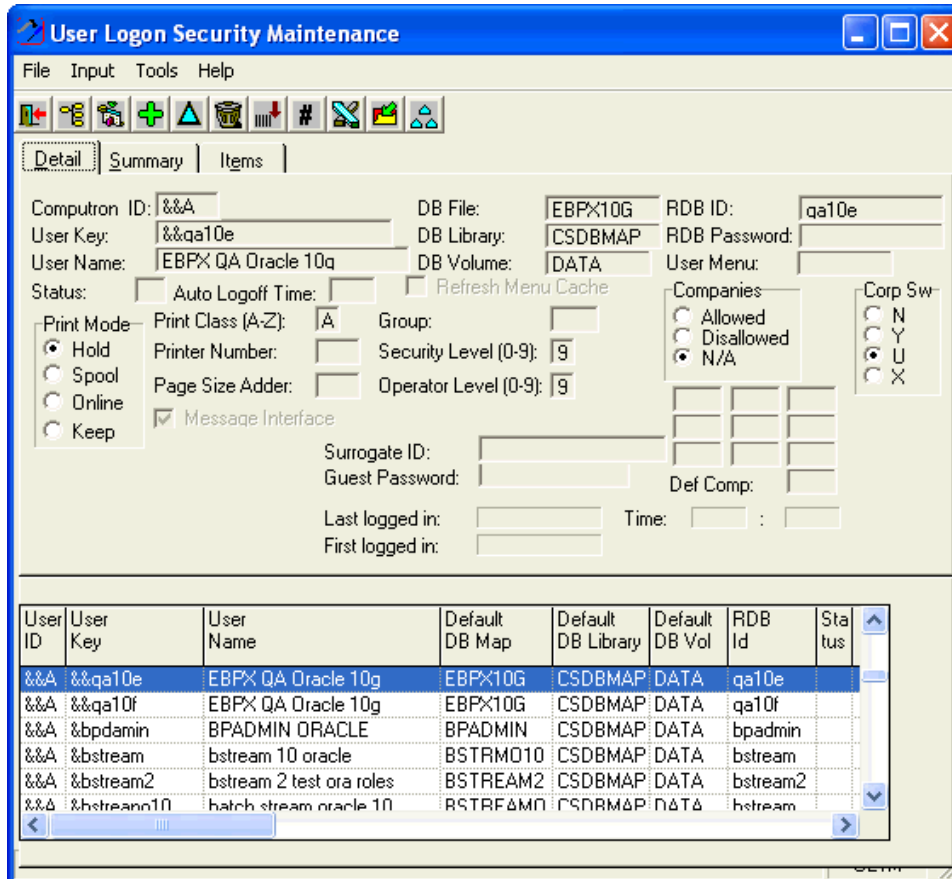
The following steps demonstrate how to add the &&A user record for a table owner:

1. On User Security Window (KS1), select the Duplicate Computron IDs check box and enter &&A in the Computron User ID field.




User Security Maintenance Window (KS1)


- Click Continue to proceed to the User Logon Security Maintenance Window (UL1).



User ID	User Key	User Name	Default DB Map	Default DB Library	Default DB Vol	RDB Id	Status
&&A	&&qa10e	EBPX QA Oracle 10g	EBPX10G	CSDBMAP	DATA	qa10e	
&&A	&&qa10f	EBPX QA Oracle 10g	EBPX10G	CSDBMAP	DATA	qa10f	
&&A	&bpdamin	BPADMIN ORACLE	BPADMIN	CSDBMAP	DATA	bpadmin	
&&A	&bstream	bstream 10 oracle	BSTRM010	CSDBMAP	DATA	bstream	
&&A	&bstream2	bstream 2 test ora roles	BSTREAM2	CSDBMAP	DATA	bstream2	
&&A	&bstream10	bstream 10 oracle	BSTRM010	CSDBMAP	DATA	bstream	

User Logon Security Maintenance Window (UL1)

- On the UL1 window, add a user with the following information:
 - ◆ In the Computron ID field, enter the value &&A.
 - ◆ In the User Key field, enter the RDBMS ID of the table owner (e.g., *CTRONSYS*).
 - ☞ Note: The User Key field entry must be unique for each table owner.
 - ◆ In the User Name field, enter the &&A value *and* the RDBMS ID that you entered in the User Key field (e.g., &&A *CTRONSYS*).
 - ◆ In the DB File field, enter a valid database map name.
 - ◆ In the DB ID field, enter the RDBMS ID of the table owner (e.g., *ctronsys*).
 - ☞ Note: The DB ID field label may display RDB ID (depending on the release of Computron software and the presentation that you are using).
 - ◆ In the Password field, enter the password that corresponds to the RDBMS ID.
 - ☞ Note: The Password field label may display RDB Password (depending on the release of Computron software and the presentation that you are using).
- Click the Add button  to add the &&A record to the User file list.





5. On the toolbar, click the Save icon  to process and save the User file.
6. Exit the User File Maintenance function and then log off the Computron desktop.
7. Reopen the Computron desktop to access the User file with your changes.


Using Generic Database IDs

RDBMS installations can take advantage of a shared &&A user record to allow multiple user ID's to share a generic table owner ID (or DB ID). This shared record can greatly reduce the amount of database administration by eliminating the need to maintain multiple database IDs and passwords across multiple User file records.

For example, your Accounts Payable personnel (*user1*, *user2*, and *user3*) can all share a DB ID called &APCLERK, provided that there's a unique &&A record named *APCLERK* in the User file with a valid DB ID and password.

The following steps demonstrate how to add a generic DB ID record for multiple user IDs.

1. On User Security Window (KS1), select the Duplicate Computron IDs check box, enter &&A in the Computron User ID field, and then click **Continue**.
2. On the User Logon Security Maintenance Window (UL1), highlight an existing &&A record (e.g., &&A *CTRONSYS*), and then click the Add icon  to access the *Add* record mode.
3. Now, add the generic &&A DB ID with the following information:
 - ◆ In the Computron ID field, accept the &&A value.
 - ◆ In the User Key field, use *all upper-case* characters to enter a generic RDBMS ID for the DB ID (e.g., *APCLERK*).
 - ☞ *Note:* The User Key field entry must be unique.
 - ◆ In the User Name field, reenter the &&A value *and* the RDBMS ID that you entered in the User Key field (e.g., &&A *APCLERK*)
 - ◆ In the DB File field, enter a valid database map name (usually *not* the table owner).
 - ◆ In the DB ID field, enter an ID for the generic database record (e.g., *apdbid*).
 - ◆ In the Password field, enter the password for the generic database record (e.g., *apdbpwd*).
4. Click the Add button  to add the record to the User file list.
5. On the toolbar, click the Save icon  to process and save the User file and return to the KS1 window.
6. On the KS1 window, enter a blank in the Computron User ID field and then click Continue. The UL1 window now lists all Computron users.
7. On the UL1 window, highlight the first user record to change and click the Change Many icon  to access the Change Many records mode. When the first record is completed, the next record is displayed in a modifiable format, allowing you to make changes.

8. In the DB ID field of each user that will refer to the shared database ID, enter an ampersand (&) followed by the table owner ID that you created in Step 3 (e.g., &APCLERK).
9. After modifying all the pertinent user records with the generic DB ID, click the Save icon  to process and save the User file and return to the KS1 window.
10. Exit the User File Maintenance function and then log off the Computron desktop.
11. Reopen the Computron desktop to access the User file with your changes.

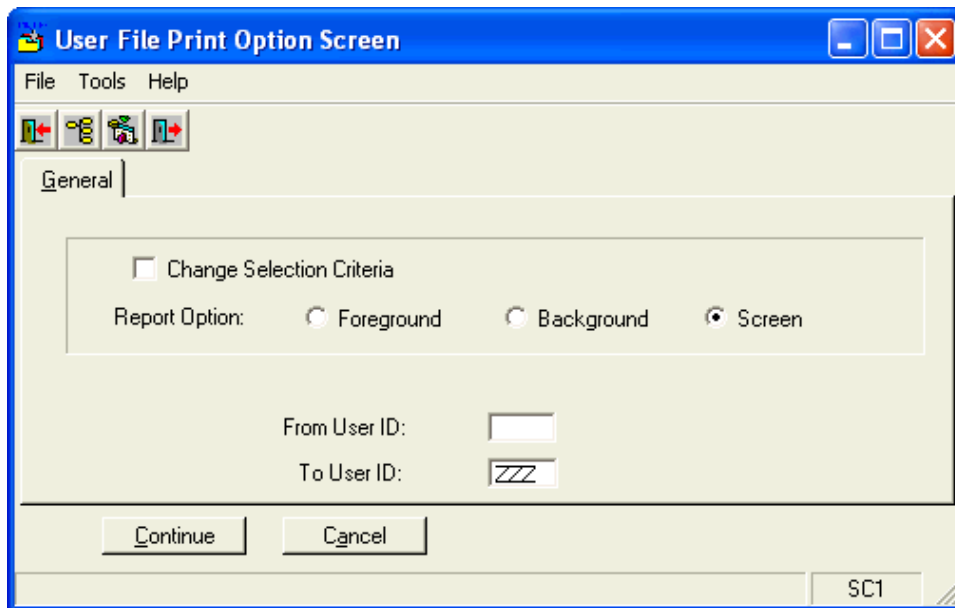
1.3 User File Inquiry/List

Introduction to User File Inquiry/List

This function is used to display a listing of the User file data established via the User File Maintenance function. This data can either be displayed online or spooled to a print file.

User File Inquiry/List Window (SC1)

The SC1 User File Print Option Screen displays when you select the User File Inquiry/List function.



User File Inquiry/List Window (SC1)

Change Selection Criteria

This field determines whether to use a query window to specify additional selection criteria. Valid options are:

- Displays the Select Query Window (SC99), which allows you to further define the selection criteria.
- Do not allow modification of the selection criteria.

Default:

Report Option

These are the processing options that produce online and printed reports. Valid options are:

Foreground – Produces a print file, processing the request in foreground. Use of this option restricts use of this window for other processing.

Background – Produces a print file, processing the request in background. Use of this option allows you to continue to process other functions from this window.

Screen – Enables you to review a list of the data online, as well as to display detail for each item on the list.

Default: Screen.

From User ID

Enter the beginning of the range of User file records to include in the inquiry/list.

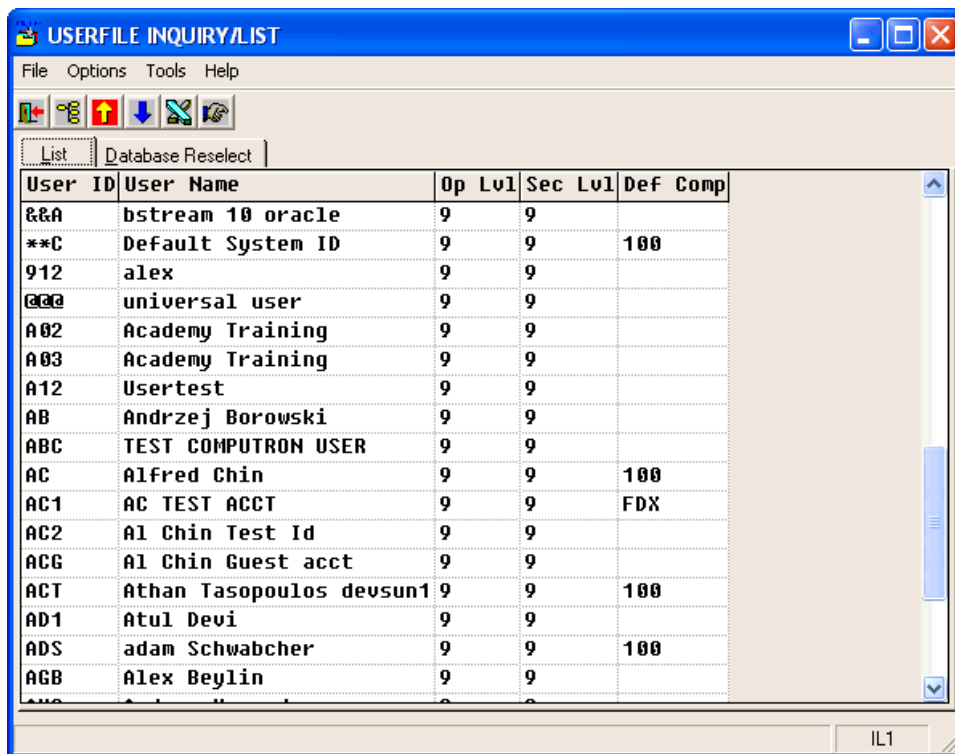
To User ID

Enter the end of the range of User file records to include in the inquiry/list.

User File Inquiry/List Report Options

Screen Option

When the Screen option is selected, an inquiry window displays:



The screenshot shows a window titled "USERFILE INQUIRY/LIST" with a menu bar (File, Options, Tools, Help) and a toolbar. Below the toolbar are two tabs: "List" (selected) and "Database Reselect". The main area contains a table with the following columns: User ID, User Name, Op Lvl, Sec Lvl, and Def Comp. The table lists various users and their attributes.

User ID	User Name	Op Lvl	Sec Lvl	Def Comp
&&A	bstream 10 oracle	9	9	
**C	Default System ID	9	9	100
912	alex	9	9	
@@@	universal user	9	9	
A02	Academy Training	9	9	
A03	Academy Training	9	9	
A12	Usertest	9	9	
AB	Andrzej Borowski	9	9	
ABC	TEST COMPUTRON USER	9	9	
AC	Alfred Chin	9	9	100
AC1	AC TEST ACCT	9	9	FDX
AC2	A1 Chin Test Id	9	9	
ACG	A1 Chin Guest acct	9	9	
ACT	Athan Tasopoulos devsun1	9	9	100
AD1	Atul Devi	9	9	
ADS	adam Schwabcher	9	9	100
AGB	Alex Beylin	9	9	

User File Inquiry/List Window (IL1)

To display the details of a listed record, you can either:

- ◆ highlight it and click the Choose/Select Item button
- ◆ double-click the desired User file record.

An inquiry detail window displays:

User File Inquiry/List Detail Window (DS1)

Print Options

The following sample report is available for the inquiry/list function:

User File Listing

```

***** Wed Apr 25, 2007 4:41 PM *
*
* USERFILE REPORT
*
*----- Page 1 *
USER USERNAME DB FILE DB LIBR DB VOL OL SL PC PM PR# CR DEF AL LOWE D/DI SALL OWED
SW COMP SW ITCH & C OMPA NIES LAST IN
-----
&&A ruser ROORA40 CSDBMAP DATA 9 9 A H 000 U
@@@ universal user ARDIINF CSDBMAP 9 9 A H 000 Y
A02 Academy Training CIGLTST0 CSDBMAP DATA 9 9 A H 000 U
A03 Academy Training CIGLTST0 CSDBMAP DATA 9 9 A H 000 U
A12 Usertest WARZTST1 CSDBMAP DATA 9 9 000 U
AB Andrzej Borowski FINAL90 CSDBMAP DATA 9 9 A H 000 U
ABC TEST COMPUTRON USER CIGLTST0 CSDBMAP DATA 9 9 A H 000 U A 100
AC Alfred Chin CIGLTST0 CSDBMAP DATA 9 9 H 001 U 100 12/12/06
AC1 AC TEST ACCT CIDMINT CSDBMAP DATA 9 9 A H 000 U FDY 12/23/03
AC2 Al Chin Test Id CIDMINT CSDBMAP DATA 9 9 A H 000 U 12/04/03
ACG Al Chin Guest acct CIDMINT CSDBMAP DATA 9 9 A H 000 Y
ACT Athan Tasopoulos devsunl WFGLTST0 CSDBMAP DATA 9 9 A H 000 U 100 02/13/07
AD1 Atul Devi FINAL10 CSDBMAP DATA 9 9 A H 000 U 02/16/07
ADS adam Schwabcher FINAL90 CSDBMAP UCDDATA 9 9 A H 000 U 100 11/30/04
AGB Alex Beylin FINAL90 CSDBMAP DATA 9 9 A H 000 U 04/26/06
AH2 Andrew Howard ARDIINF CSDBMAP DATA 9 9 A H 000 U 05/04/06
AHB Andrew Byrne FINAL90 CSDBMAP DATA 9 9 A H 004 U 100 10/11/05

```

1.4 Menu File Maintenance

Introduction to Menu File Maintenance

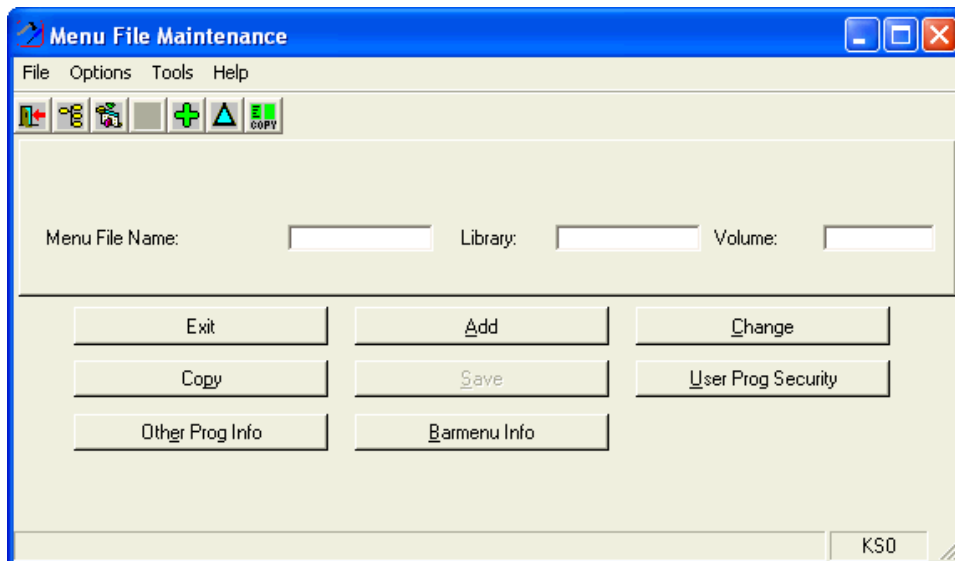
Computron menus are defined and maintained via the Menu File Maintenance (MENUMNT2) function. The Menu File Definition windows allow you to change the format of any Computron menu. You can easily change a selection on the menu by editing the appropriate line item.

Each Computron menu is assigned basic parameters and security parameters. Basic parameters define the program, procedure, or menu that is run when a menu option is selected. Basic parameters also define the parameters passed from the menu to a specific process. These passed parameters are maintained via the *Other Info* button, which accesses the Other Program Information Windows (DS5) or (DS9). Security parameters can be used to restrict access to menu selections (used in conjunction with the Computron User ID and Security Level fields in the User file) In addition you can establish a password that must be entered by the user before the menu selection can be accessed.

Once the changes are saved, they are immediately reflected in the system. The menu is propagated to the local cache. The programs or procedures that actually display the menu or activate its selections are unaffected.

Menu File Maintenance Window (KS0)

The Menu File Maintenance Window (KS0), which is the first window to display, allows you to access the Menu File Maintenance options. To proceed to a particular maintenance option, you must first enter the name of the Menu file to maintain.



Menu File Maintenance Window (KS0)

Menu File Name / Library / Volume

This is the physical location (file/library/volume) of the Menu file to maintain.

☞ *Note:* Computron’s standard name for menu file names consists of a prefix identifying the particular system (e.g., GL, AP, AR, UV) followed by a maximum of six characters, which indicate program functionality. For example, the menu file name for Universal Utilities is UVUTIL. Computron’s standard name for menu libraries consists of a prefix identifying the particular system (e.g., CIUV, CIGL, CIAP, CIAR) followed by MENU. For example, the library where Universal Utilities menus are contained is named CIUVMENU.

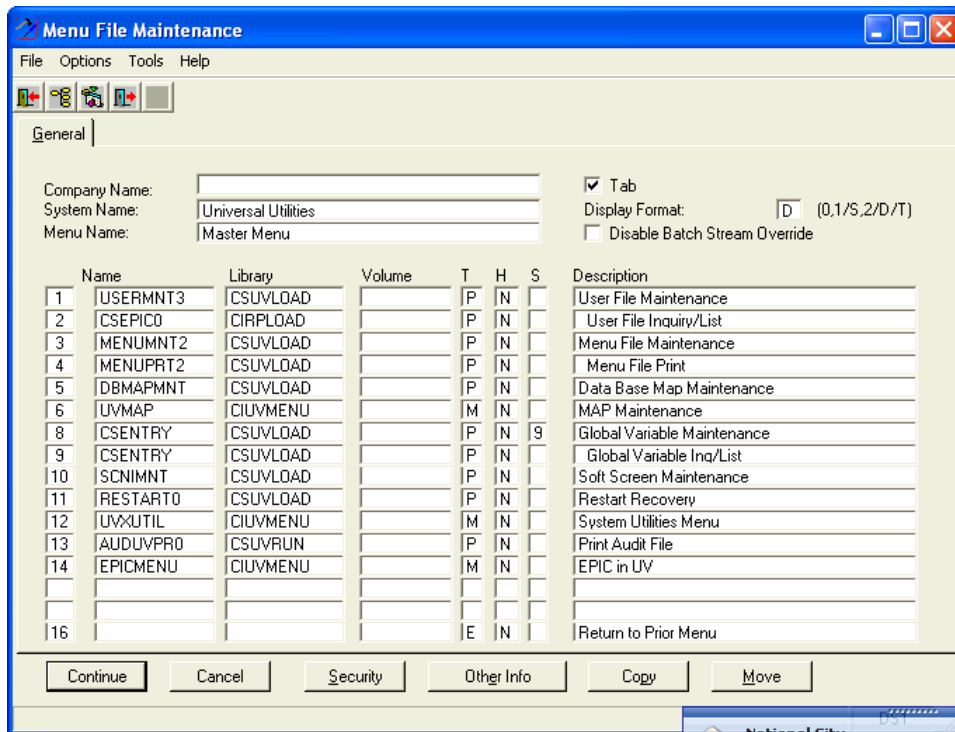
Table 1.4-1 Options on the Menu File Maintenance Window (KS0)	
Button	Function
Exit	<p>Exit the Menu File Maintenance function.</p> <p>☞ <i>Note:</i> If you exit without clicking Save, none of the changes or additions made during the current session are saved.</p>
Add	<p>Add a new Menu file. This displays the Menu File Definition Window (DS1) where you can begin the design of the menu. For example, from this window you can indicate the menu selections for the menu and which users are allowed/disallowed access to the menu selections.</p> <p>If a Menu file has not been saved, you can use the Add button to edit the Menu file before saving it.</p> <p>☞ <i>Note:</i> It is strongly recommended that you copy an existing menu file to create a new one. By copying, you are including all the controls required for the menu.</p>
Change	<p>Change an existing menu file. This displays the Menu File Definition Window (DS1) where you can make changes to the design of the menu, and the security privileges for each menu selection.</p>
Copy	<p>Copy an existing Menu file.</p>
Save	<p>The Save option is inactive or missing from the Menu File Maintenance Window (KS0) until after a menu has been modified.</p> <p>Save any of the changes or additions that have been made during this session.</p> <p>☞ <i>Note:</i> If you exit Menu File Maintenance before clicking Save, any menus modified or added during the session are not saved. If you are saving changes made to an existing menu, the following error message window displays:</p> <p style="margin-left: 20px;">File Already Exists. Override To Scratch Existing File.</p> <p>Click the OK button to save the menu changes.</p>

Table 1.4-1 Options on the Menu File Maintenance Window (KS0)	
Button	Function
User Prog Security	<p>Access the Security Maintenance Window — Menu File Security Maintenance Window (DS4). This window offers you a quick method of establishing security privileges for the Menu file.</p> <p>Typically, security privileges are entered from the Menu File Definition Window (DS1) on a function-by-function basis; i.e., for each function listed as a menu selection. The Menu File Security Maintenance Window (DS4), however, allows you to establish general security privileges without accessing specific menu selections; e.g., from this window you can disallow a user from accessing any function on the menu.</p>
Other Prog Info	<p>Access the Other Program Information Window (DS5) where you can enter additional information about each menu selection accessed from the menu. For example, you can indicate the function version number for reference purposes or establish certain parameters to automatically pass to the function whenever it is accessed from the menu.</p>
Barmenu Info	<p>Access the Barmenu Information Window (DS16) where you can modify menu bar and icon information for functions that are accessed from the menu. This type of information is only intended for AUI implementations. The Barmenu is only available once the menu has been displayed in Change mode.</p>

Menu File Definition Windows (DS1 and DS2)

The Menu File Definition function actually consists of two related windows: the DS1 and DS2. The Menu File Maintenance Window (DS1) represents the first step in establishing a menu. It displays when you elect to modify an existing menu or add a new menu.

This window allows you to define the menu and to establish for each selection on the menu security parameters such as a program password that must be entered by the user to access the selection. These features and others are discussed in the sections that follow.



Menu File Maintenance Window (DS1)

Company Name

This is the optional name of the company. The company name can be up to thirty alphanumeric characters. If left blank, the system uses the company name from the previous menu at run time. The company name displays in the first header line above the menu screens used by Computron's character cell presentation.

System Name

This is the name of the system for which the menu is being created (e.g., Accounts Payable, General Ledger). The system name can be up to thirty alphanumeric characters. If left blank, the system uses the system name from the previous menu at run time. The system name displays in the second header line above the menu screens used by Computron's character cell presentation.

Menu Name

This is the name of the menu together with the version number of the menu (e.g., *Universal Utilities*). The menu name can be up to thirty alphanumeric characters. If left blank, the system uses the menu name from the previous menu at run time. This name is the fourth header line above the menu screens used by Computron's character cell presentation. It displays after the header line that indicates your database map (e.g., CIGLTST0 in CSDBMAP on DATA).

Tab

This option is only used for Computron's character cell presentation.

This field indicates whether a Tab character is displayed on the menu window for each program entry. Valid options are:

- Tab characters are displayed on the menu window for each function key selection. The user may tab from one tab character to another and, with the cursor positioned on the tab character for the desired selection, press Enter to access the selection. Or, if multi-workstation is loaded, the entire field is highlighted and the user can go from one entry to the next by entering the first letter of the program description. Additionally, the user can press the corresponding function key, without positioning the cursor on the Tab character.
- Tab characters do not appear next to function key selections on the menu window. The user must select the desired option by pressing the corresponding function key.

Display Format

This option is only used for Computron's character cell presentation.

Each menu has certain formatting features in common, including the layout of menu selections and whether they are subdivided into one or two columns. Each selection includes the function key that allows access, as well as a description of the function. The Display Format function allows the user to customize the appearance of these menu selections within the standard format by indicating:

- ◆ the number of columns the user wants the menu divided into (one or two columns);
- ◆ whether the entries under each column should be printed without any spaces between them;
- ◆ subdivided into groups of four with a blank line between each group. For example, for a one-column menu with six entries, this feature would print the first four functions in one group and the remaining two in the other.

This option is only used for Computron's character cell presentation.

Keep the following considerations in mind when designing the Display Format of a menu:

- ◆ a one-column menu can contain a maximum of ten lines, and:
 - the sixteenth line of the first Menu Definition window (which is usually reserved for the Exit key), and
 - the enter (or PF22 prompt).
- ◆ a two-column menu can contain a maximum of sixteen lines, and:
 - the Enter (or PF22 prompt).

Therefore, if the user is designing a menu with twelve entries, they would need two columns.

The following table describes the available options for the Display Format field and the maximum number of lines per column for each option.

Option	Description	Max. Entries Per Column
1	<p>Creates a one-column menu with selections that are divided into two groups of four. The groups are separated by one blank line.</p> <p><i>Note:</i> The maximum does not include the sixteenth line entry on the first Menu File Maintenance window (usually reserved for the Exit function) and the Enter (or PF22) prompt.</p>	8
S	<p>Creates a one-column menu without dividing the selections into separate groups. The user can instead enter blank line items as part of the menu to create groupings of selections.</p> <p><i>Note:</i> The maximum does not include the sixteenth line entry on the first Menu File Maintenance window (usually reserved for the Exit function) and the Enter (or PF22) prompt.</p>	10
2	<p>Creates a two-column menu with the selections divided into four groups of four. The two groups in each column are separated by one blank line.</p> <p><i>Note:</i> The maximum does not include the Enter (or PF22) prompt.</p>	8
D	<p>Creates a two-column menu without dividing the selections into separate groups. The user can instead enter blank line items as part of the menu to create groupings of selections</p> <p><i>Note:</i> The maximum does not include the Enter (or PF22) prompt.</p>	8

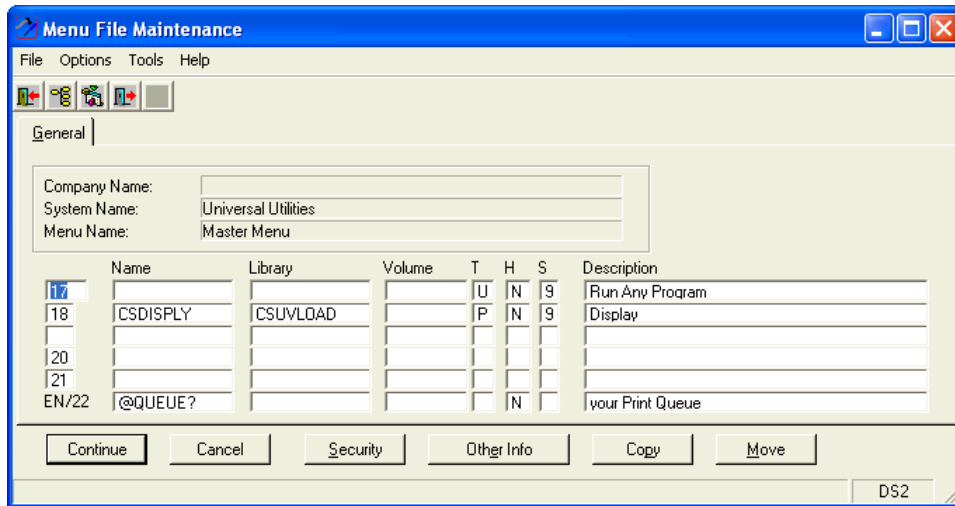
Option	Description	Max. Entries Per Column
T	<p>Creates a two-column menu that displays up to twenty selections: ten selections in the first column and up to ten selections in the second, until it encounters a selection type of E (Exit) or L (Logoff). For further details, refer to the T (Type) field description later in this section.</p> <p>Therefore, the first four entries on the Menu File Definition Window (DS2), which are typically hidden, actually display on the menu (i.e., if the last entry is the only selection that allows you to exit or logoff).</p> <p><i>Note:</i> The maximum does not include the Enter (or PF22) prompt.</p>	10

Disable Batch Stream Override

This field controls batch stream processing. You can access many Computron application functions that have been enabled for processing via Job Processing (a.k.a. Batch Stream Processing) without making any permanent changes to the menus. Use this field to inhibit users from temporarily overriding the entry in the B field on the Other Program Information Window (DS5). Valid options are:

- Inhibits users from overriding the value that indicates that the function is processed as a part of Job Processing.
- Allows users to override the Job Processing values for functions on the menu.

The second Menu File Maintenance Window (DS2) is accessed by clicking Continue on the DS1 window.



Menu File Maintenance Window (DS2)

The second window is similar in content to the first window; however, it serves a slightly different purpose.

The first window lists sixteen line items — each corresponding to an available selection on the menu. This line item can either be a header (descriptive information only), a program or a procedure that can be accessed from the menu.

The second window also contains line items. Unlike those on the first window, however, the first five line items on this window are designed to contain hidden entries, i.e., functions that can be accessed by the user although they are not displayed on the menu. The last line item is reserved for the Enter key.

Each of the line items that follow corresponds to an available selection on the menu. The sixteen line items from the first Menu File Definition Window (DS1) are designed for those selections that display on the menu. The remaining line items on the second Menu File Definition Window (DS2) with the exception of the line hard-coded as EN/22 are designed for those hidden functions that can be accessed by the user but which do not display.

The fields that constitute each line item are as follows:

Function Key

This option is only used for Computron's character cell presentation.

The first two positions of each line item, except for those in the last line item, contain the function keys that enable access to the menu selections in the character cell presentation. The last line item is reserved for the Enter key.

Default function keys are displayed when you first access the Menu File Maintenance windows (PF1 to PF16 on the first window, and PF17 to PF21 on the second window). These defaults are displayed for reference purposes only and can be modified as necessary.

- ◆ Be sure to enter function keys in sequential order so that they are displayed similarly on the menu screens used by Computron's character cell presentation. In addition, it is

important to consider the function key conventions that have been developed for the installation. For Computron menus, F21 and above are reserved for Computron functions.

- ◆ F28 to F32 are automatically activated by the Menu File Maintenance function as follows. You can either choose to highlight these functions by including them as a line item on the menu or leave them hidden.

Table 1.4-2 Line Items — Menu File Maintenance Window (DS2)		
Function Key	Button	Description
28	Reset To Original (Button)	From any Computron menu, this allows the user to automatically restore the default database map established for the user via User File Maintenance. This option is normally used only after the current database map has been previously modified (see Function Key 29 below). This option is available to a user only if the Corp field in the corresponding User file record has been set to U or X.
29	Change Database Map	Allows the user to access the Change Database window where Database Maps can be switched. This option is available to a user only if the Corp field in the corresponding User file record has been set to U or X.
30	Escape	Returns the Computron user to the Computron Main Menu, (e.g., the first menu to appear after the user logon)
31		Takes the user to the soft screen version (Set Defaults) of the current window.
32	Exit	Logs the user off the system from the menu.

☞ *Note:* To print a comment or header on the menu, leave the function key blank. For further details on entering a comment line, refer to the description of the T field.

Name / Library / Volume

This is the physical location (name/library/volume) of the program, procedure or lower-level menu to be accessed at this particular menu selection. In addition, if the menu selection is used to switch database pointers, these fields contain the location of the new database map.

For each type of allowable entry (program, procedure, or lower-level menu), the following table lists the standard library where it resides. Each library name contains a four-letter prefix that identifies the system to which it pertains (CIGL for General Ledger, CIAP for Accounts Payable, etc.). This prefix is followed by another identifier that indicates the type of entries the library contains (e.g., LOAD for programs).

Types of Entries Allowed in the Name Field	
Allowable Entry	Standard Library

Program	System LOAD Library (e.g., CIGLLOAD for Computron's General Ledger system).
Shell Script	System RUN Library (e.g., CIAPRUN for Computron's Accounts Payable system).
Menu	System MENU Library.
Switch To a New Database Map	Physical location of the new map (e.g., CSDBMAP).

If the volume field is left blank, it automatically defaults to the volume where the current menu resides. If the Library field indicates that the menu selection is used to effect a database map change, then the system uses the Database Map volume. If the Volume field contains a special string, the system performs a specific substitution. The following table lists the special strings and associated substitutions:

Special String	Substitution
&IN	Currently defined INVOL.
&OUT	Currently defined OUTVOL.
&RUN	Currently defined RUNVOL.
&WORK	Currently defined WORKVOL.
&CTRON	Currently defined CTRONVOL.
&DBMAP	Currently defined DBMAPVOL.
&IPL	Currently defined IPLVOL.
&SPOOL	Currently defined SPOOLVOL.

T

This field identifies the nature of the menu selection or indicates the type of function to be performed by the menu selection. Valid entries are:

B – A procedure with this function is submitted to a background task for subsequent processing. The terminal is therefore available to the user for other functions while the procedure is being processed.

D – The menu selection displays the Change Database Map window where the user is prompted for the volume/library/file of the alternate database map to temporarily override the database map established for the user in the User file. This override is in effect until the user exits from Computron's Main menu; then, the original database map established in the User file is restored. This menu selection serves the same function as the Change Database Map icon. This option is particularly useful for those users whose User file Corp field value is N or Y (indicating that they cannot modify their database map directly).

E – This value allows the user to exit the menu and return to a higher-level menu. Computron's standard Escape icon serves the same function.

Example: Suppose the user has accessed the System Administrator menu from the General Ledger Master menu. An entry of this type in the System Administrator menu returns the user to the General Ledger Master menu.

G/g – This identifies a menu selection to suppress when displaying under the AUI presentation.

L/l – This identifies a program to run on the client machine (local PC). The program can be an executable (.exe) or an associated executable program type. There is an eight-character limit for the filename – including the extension. If no extension is provided, it is assumed to be .exe. The volume is converted to a DOS path using the local LGMAP.DAT file; then the library and file name are appended. If the file cannot be launched via this absolute pathname, the filename is launched without a path – relying on the standard DOS path mechanism.

☞ *Note:* Additional command line arguments can be added to the program via the Other Information Window (DS9L) for local programs. For detailed instructions on accessing and using this window, refer to the Other Program Information windows later in this section.

M – This identifies a lower-level Computron menu that can be accessed from this menu. The Name/Library/Volume fields contain the name and location of the menu.

P – This identifies a Computron program or command procedure that requires information from the User file (such as security level data) before it can run.

p – This identifies a program or command procedure that does not require information from Computron's User file.

☞ *Note:* If the p option is used for a Computron program that requires information from the User file and the user has a security level of nine, the system restores the original User file information for that program. Otherwise, it halts execution of the program with the following error message:

```
"User ID: XXX – You Can Not Access This System Module At This
Time – Reason: User XXX May Not Access This Program".
```

U – The menu selection displays the Run Any Program window, where the user is prompted for the volume/library/file of the program to be executed. Any program the user runs from this window is supplied with parameters from the User file. This menu selection serves the same function as the Run Any Program icon and is generally used in conjunction with security level restrictions.

u – The menu selection displays the Run Any Program window, where the user is prompted for the volume/library/file of the program to be executed. Any program the user runs from this window is **not** supplied with parameters from the User file. Therefore, this menu selection is best suited to running system functions rather than Computron functions.

X – Identifies a comment or header. The only other entry required for a header or comment line within the Menu File Maintenance Window (DS1) is the literal to display on the window, which is entered in the Description field.

H

This field indicates whether the program can be terminated with Ctrl+C while running under the character cell presentation. Valid entries are:

Y – Disables Ctrl+C during program execution. It is only valid when a program or procedure is run (G, L, l, P, p, S, U, u).

N – Enables Ctrl+C during program execution.

S

This field, which is used in conjunction with the Security Level field in the User file, allows you to control access to menu selections.

The security level zero to nine (0-9) established here is compared to the corresponding Security Level field in the User file. If the security level for this menu selection is greater than the security level in the User file, the user is unable to access this selection on the menu.

☞ *Note:* For Computron’s character cell presentation, when a user is denied access to a selection, the Display Format field determines whether the menu selection is still displayed on the user’s menu. If the Display Format field is one or two, the selection is not displayed.

Description

The description of the selection as it appears on the menu.

☞ *Note:* Computron recommends that you do not use special characters in this name. Single quotes, double quotes, etc. can create errors when running scripts.

☞ *Note:* Any description entered for the selections on the DS2 window does not display on the menu.

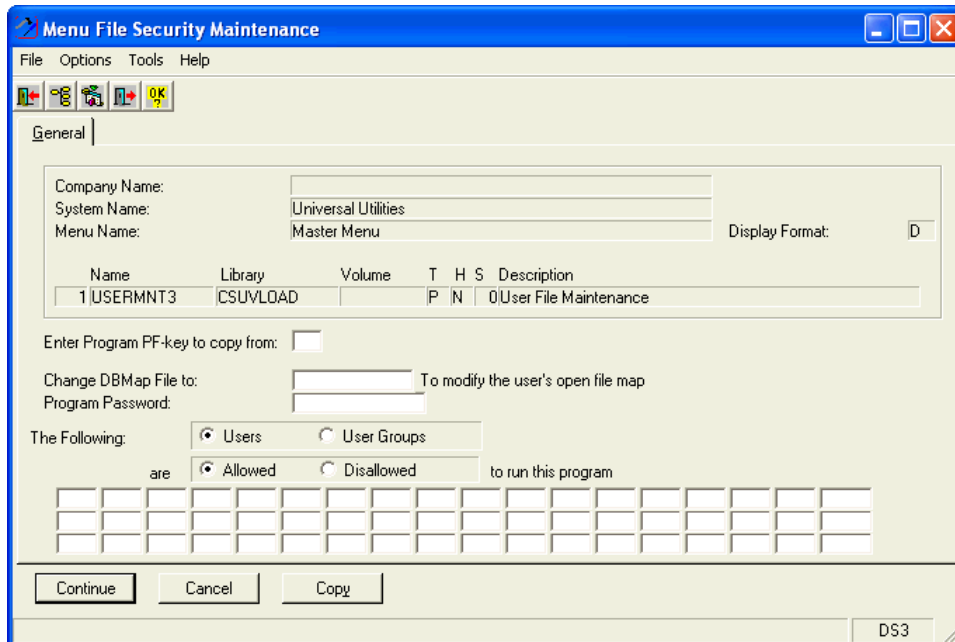
Once you have entered text on this window, you can easily manipulate the text using the Move and Copy buttons as described in the following table. Before clicking the Copy or Move buttons, be sure to examine the line item to which the text is being moved or copied. Any existing data on the line item is overwritten when the Copy or Move button is clicked.

Table 1.4-3 Options on the Menu File Definition Windows (DS1 & DS2)	
Button	Function
Continue	From DS1 — Proceed to the Menu File Definition Window (DS2). From DS2 — Proceed to the Menu File Maintenance Window (KS0).
Cancel	Return to the previous window.
Security	Access the Line Item Security Maintenance Window (DS3) where you can establish security parameters for the selected line item. Before clicking this button, position the cursor on the line item for which security is being established.

<p>Other Info</p>	<p>Access the Other Program Info Maintenance Window (DS9) where you can establish additional parameters to pass to the selected program for the line item. Before clicking this button, position the cursor on the line item for which parameters are to be established.</p> <p><i>Note:</i> If the cursor is positioned on a line item with a T field value of 1, the Other Program Info Maintenance Window (DS9) is accessed when the Other Info button is selected.</p>
<p>Copy</p>	<p>Copy the selected line item. An additional window displays, allowing you to indicate the location to which the line item selected is to be copied.</p>
<p>Move</p>	<p>Move the selected line item. An additional window displays, allowing you to indicate the location to which the line item selected is to be moved.</p>

Menu File Line Item Security Window (DS3)

The Line Item Security Maintenance Window (DS3) allows you to establish security parameters for each selection on the menu. This window can be accessed from the Menu File Definition Window (DS1) or (DS2) by positioning the cursor on the desired selection (line item) and clicking the Security button.



Line Item Security Maintenance Window (DS3)

Enter Program PF-Key to copy from

Use this field to enter another line item (identified by function key) on the same menu from which the security options are to be copied. If this field is entered, you must click the Copy button to execute the copy operation.

The following security parameters can be entered directly or modified following the completion of the copy operation.

Change DBMap File to

Use this field to change the Database Map file currently in use when the line item is selected from the menu. When the user exits the function, the Database Map file is changed back to the original (default) database map for the user. For example, this can be utilized when you want to set up an initial menu with menu selections Live GL and Test GL.

Program Password

Use this field to force the user to enter a password prior to execution of the selected line item. Up to nine characters can be specified as the password. When this option is used, a prompt for password displays on an additional window prior to program execution or lower-level menu display. The user must enter the correct password to continue.

The Following Users /User Groups are Allowed/Disallowed to run this program

This series of fields are used to specify individual users (Computron user IDs), user groups or workstations that are to be restricted from or allowed access to this menu selection.

First, indicate whether:

- ◆ Users are to be allowed/disallowed access to the menu selection
- ◆ User Groups are to be allowed/disallowed access to the menu selection

Then, indicate whether these entities are to be:

- ◆ Allowed access to the menu selection
- ◆ Disallowed access to the menu selection.

The remaining fields are used to enter the users (Computron user IDs), or user groups or workstations (by workstation number) to which the allowance/restriction applies. Up to fifty-four users, user groups or workstations can be entered in these fields.

Table 1.4-4 Options on the Line Item Security Maintenance Window (DS3)	
Button	Function
Continue	Proceed to Menu File Definition Window (DS1).
Cancel	Exit Line Item Security Maintenance Window (DS3) without retaining changes.
Copy	Copy the security options of the line item indicated by the value entered for the Enter Program PF-Key to copy from field to the menu selection under maintenance.

Menu File Security Maintenance Window (DS4)

To modify security on the Security Maintenance Window (DS4), enter the menu name and location on the KS0 window, click the Change button, Cancel from the DS1 window and then click the User Prog Security button. It offers you a quick method of establishing security privileges for a particular user without accessing individual line items on menus. Typically, security privileges are entered from the Menu File Definition Window (DS1) on a function-by-function basis; that is, for each line item listed as a selection on the menu.

Example: To restrict a user named *USR* from accessing every selection on the menu without this window, you would have to follow these steps:

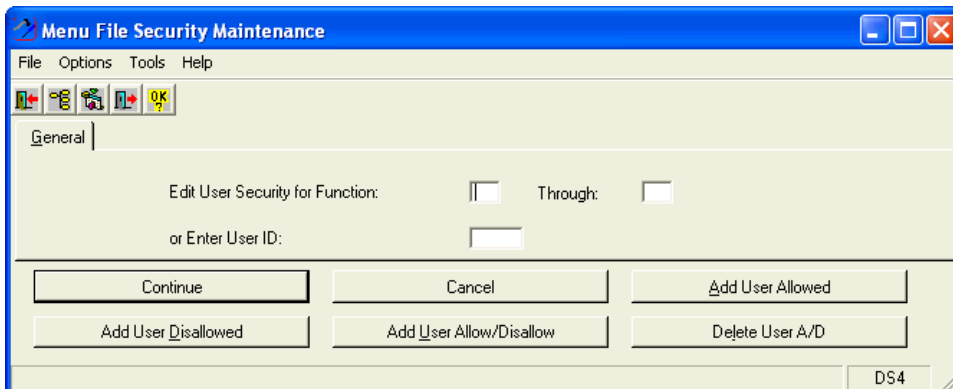
1. Access the Line Item Security Maintenance Window (DS3) and select:

The Following Users Are Disallowed To Run This Program.

2. Add *USR* to the list of Computron users to which this restriction applies.
3. Repeat steps 1 and 2 for every line item on the Menu Definition window.

The Security Maintenance Window (DS4) simplifies the process by allowing you to enter security privileges that pertain to a range of line items. In other words, you can allow/disallow a user from

accessing a particular menu selection based on the existing Allowed/Disallowed field on the corresponding Line Item Security Maintenance window. Check to make sure that *Users* is selected on the Line Item Security Maintenance window for each line item that you want to affect.



Security Maintenance Window (DS4)

Example: Provided the allowance/restriction applies to *Users* and not to *User Groups* or *Workstation* you can prevent users from accessing those selections for which the Allowed/Disallowed field is set to Disallowed.

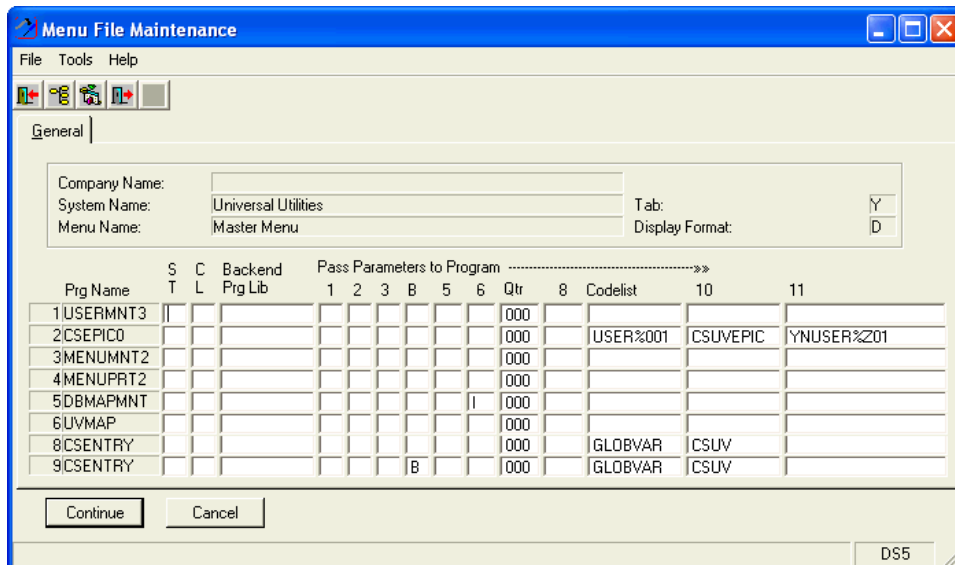
Table 1.4-5 Options for the Security Maintenance Screen (DS4)	
Button	Function
Continue	<p>Access the Line Item Security Maintenance Window (DS3) for the first line item in the range of function keys entered.</p> <p>Each time the Continue button is selected, the Line Item Security Maintenance Window (DS3) for the next line item in the range entered displays. Once the last line item for the range of function keys is entered you return to the Menu File Maintenance Window (KS0).</p>
Cancel	<p>Exit the function without saving the security options you have entered during this session and return to the Menu File Maintenance Window (KS0).</p>
Add User Allowed	<p>Allow the Computron user to access the range of function keys indicated above, i.e., all of the function keys in the range with the selection:</p> <p style="padding-left: 20px;">"The Following Users Are Allowed To Run This Program"</p> <p>The Computron user ID is added to the list of users allowed on the appropriate Line Item Security windows.</p> <p><i>Note:</i> If no range is entered, the user is permitted to access all line items with the selection:</p> <p style="padding-left: 20px;">"The Following Users Are Allowed To Run This Program"</p>

Table 1.4-5 Options for the Security Maintenance Screen (DS4)	
Button	Function
Add User Disallowed	<p>Disallow the Computron user from accessing the range of function keys indicated above, i.e., all of the function keys in the range with the selection:</p> <p style="padding-left: 20px;">"The Following Users Are Disallowed To Run This Program"</p> <p>The Computron user ID is added to the list of users disallowed on the appropriate Line Item Security windows.</p> <p>☞ <i>Note:</i> If no range was entered, the user is denied access to all line items with the selection:</p> <p style="padding-left: 20px;">"The Following Users Are Disallowed To Run This Program"</p>
Add User Allow / Disallow	<p>Allow or Disallow the Computron user from accessing the range of function keys indicated above based upon the value of the Allowed/Disallowed field for each entry. In other words, the user is:</p> <ul style="list-style-type: none"> ◆ denied access to line items with the selection: <li style="padding-left: 20px;">"The Following Users Are Disallowed To Run This Program" ◆ allowed access to those line items with the selection: <li style="padding-left: 20px;">"The Following Users Are Allowed To Run This Program" <p>The Computron user ID is added to the list of Allowed/Disallowed Computron user IDs on the appropriate Line Item Security windows.</p> <p>☞ <i>Note:</i> If no range is entered, this entry pertains to all selections on the menu.</p>
Delete User A/D	<p>Delete the user from the list of Computron user IDs for each of the function keys in the range indicated above.</p> <p>The Computron user ID is deleted from the list of Users Allowed/Disallowed on the corresponding Line Item Security windows.</p> <p>☞ <i>Note:</i> If no range is entered, this entry pertains to all selections on the user menu.</p>

Other Program Information Windows (DS5)

To modify information on the Other Program Information Window (DS5), enter the menu name and location on the KS0 window, click the Change button, Cancel from the DS1 window and then click the Other Program Info button on the Menu File Maintenance Window (KS0). It allows you to enter additional information about each menu selection.

Example: You can indicate whether the back-end library should be used for subsequent links of the program to that new library. In addition, you can enter several types of parameters to pass to the selection when it is run.



Other Program Information Window (DS5)

ST

This option is not available.

CL

This option is not available.

Backend Prg Lib

Some Computron programs run in modules using a front-end and a back-end process. The name of the backend process for a program can be modified here.

The remaining fields on this window are reserved for specifying parameters to pass to the menu selection when it is run. They are as follows:

1 2 3

Reserved for use by specific programs.

B

This parameter is applicable only if the line item is a function that is processed as a part of Batch Stream Processing. See Chapter 3, “Batch Stream Processing,” for additional information. Valid entries are:

M – This value indicates that Computron users are required to make entries on the Batch Stream Maintenance Window (BS1) before the function can run. This window prompts the user for information that the Batch Stream Processor uses to run the function (e.g., the selection criteria used to select records for reporting).

Blank – This value indicates that the Batch Stream Maintenance Window (BS1) is not displayed to users; the users proceed directly to the other windows in the function where they can select options and initiate function processing as usual.

B – This value indicates that the Batch Stream Maintenance Window (BS1) is displayed to users. However, by clicking Continue, users can bypass this window and continue to the other windows in the function.

5, 6, QTR, 8

These fields are reserved for use by specific programs.

Codelist

This parameter is used in conjunction with any Code List Maintenance function — CSENTRY program. CSENTRY is the Code Dictionary Maintenance program that maintains the standard Computron code list tables. If only one code list is maintained for this particular function, you can identify the name of this code list by entering that name in the Codelist field to bypass the first window in the function that prompts for a code list name.

☞ *Note:* When the program name is CSEPIC0, this field has a special use. The entry in this field is the Soft Screen program name. The program name for standard Computron reports is the same as the report ID. If both a print and online version of a report is available, the report ID for the printed version is used in this field.

10

For CSEPIC0, this field is used to specify the logical library name as it is found in the database map for the EPIC report library. The standard naming convention used by Computron is: XXXXEPIC, where XXXX is the system prefix (CSUV for Universal Utilities, CIGL for General Ledger, etc.).

11

In conjunction with the CSEPIC0 program, entries are in the format:

XYCCCC%Z01

Where the first position (X) is used to indicate whether the Inquiry/List Window (SC1) prompts the Computron user to modify the report's record selection. If Y is entered in this field, the Inquiry/List Window (SC1) prompts the user to enter a value in the Change Selection Criteria field.

The second position (Y) is used to determine whether the Inquiry/List Window (SC1) prompts the user for a report ID. If Y is entered in this field, the Inquiry/List Window (SC1) prompts the user to enter a value in the Report ID field at run time. The user is then allowed to use the Q-^{Mark} facility to select an EPIC report.

☞ *Note:* Only the last four positions of the field are Q-Markable. As such, all alternative report definitions must be established with the same first four characters. If only one version of the report is available, this field is set to N.

Positions three through ten (CCCC%Z01) are used to enter the report or zoom ID. If a zoom type report ID is entered, the run-time prompt window allows the user to run the report on the window. If the zoom report ID contains a valid report ID in the Associated Report field of the report definition, the user is also allowed to run the report in either Foreground or

Background mode. If the Associated Report field is left blank in the zoom report definition, only the window option is valid at run time.

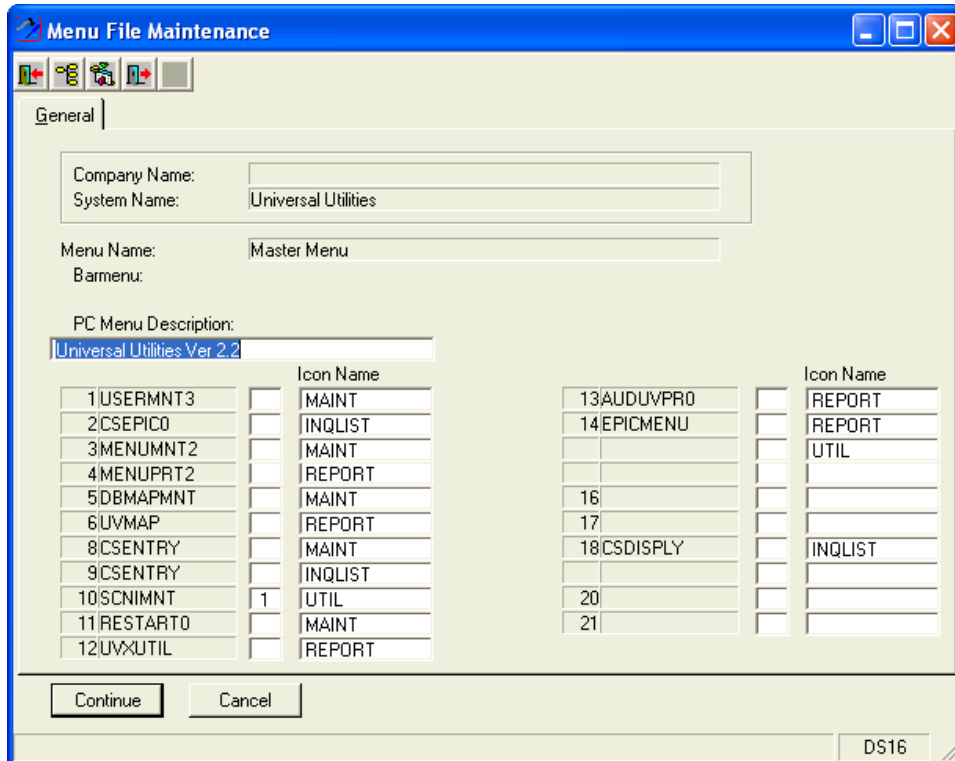
If no entry is made in positions three through ten, the system refers to the Codelist field and uses the soft screen program name as the report ID.

As a standard, printed reports are defined ending with the characters 001, 002, 003, etc. Alternatively, zoom reports are defined ending with the characters Z01, Z02, Z03, etc..

Other program information can also be entered for a single menu selection by clicking the **Other Info** button from the Menu File Definition Windows (DS1) or (DS2). This allows you to enter the fields described on the previous pages for a single line item.

Barmenu Information Window

This window allows you to modify the icons that represent menu items. This function is commonly used to remove program icons from menus that the user does not have access to. To modify an icon, enter the menu name and location on the KS0 window, click the Change button, Cancel from the DS1 window and then click the Barmenu Info button.



		Icon Name		Icon Name			
1	<input type="checkbox"/>	USERMNT3	MAINT	13	AUDUVPRO	<input type="checkbox"/>	REPORT
2	<input type="checkbox"/>	CSEPIC0	INQLIST	14	EPICMENU	<input type="checkbox"/>	REPORT
3	<input type="checkbox"/>	MENUMNT2	MAINT			<input type="checkbox"/>	UTIL
4	<input type="checkbox"/>	MENUPRT2	REPORT			<input type="checkbox"/>	
5	<input type="checkbox"/>	DBMAPMNT	MAINT	16		<input type="checkbox"/>	
6	<input type="checkbox"/>	UVMAP	REPORT	17		<input type="checkbox"/>	
8	<input type="checkbox"/>	CSEENTRY	MAINT	18	CSDISPLY	<input type="checkbox"/>	INQLIST
9	<input type="checkbox"/>	CSEENTRY	INQLIST			<input type="checkbox"/>	
10	<input type="checkbox"/>	SCNIMNT	1	UTIL	20	<input type="checkbox"/>	
11	<input type="checkbox"/>	RESTART0	MAINT			<input type="checkbox"/>	
12	<input type="checkbox"/>	UVXUTIL	REPORT	21		<input type="checkbox"/>	

Barmenu Info Window (DS16)

PC Menu Description

Use this box to enter a description for the menu that you are modifying. The description appears in the window's title bar.

Icon Name

The default program and menu icons (.ico) are released with Computron Software and are stored either on the local PC or in a shared network location, as follows:

- ◆ For AUI Visual Basic desktops, the program icons are stored in the directory:
`./.../exec/_ctron_/pics.`

To remove an icon, simply delete its corresponding name from the Icon Name column. To access a different icon, enter the icon file name in the Icon Name column. Click Continue to return to the Menu File Maintenance Window (KS0) and click Save to save your changes.

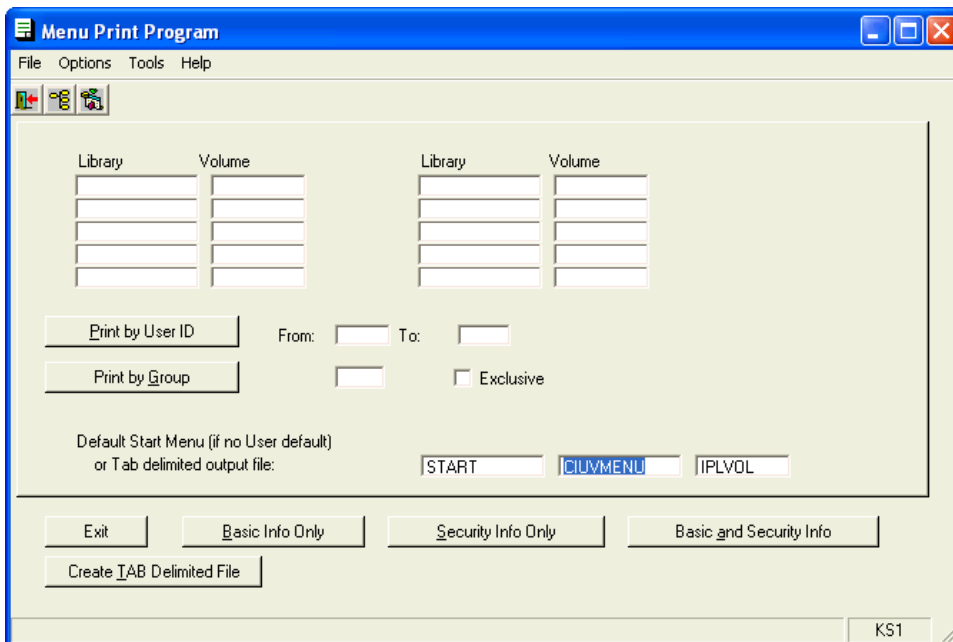
☞ *Note:* When replacing icons, make sure to copy your replacement icon files into the appropriate location before accessing this function.

Tailoring Menus

For all Computron application systems, standard menus are released as a part of the installation process. However, you may need to tailor these menus to incorporate security or other options at your site. The following pages illustrate (using standard menus from the Computron Fixed Assets system) how you accomplish this tailoring process.

Note: This is only an illustration. The security options described are for illustrative purposes only and are not to be interpreted as recommendations. You should plan changes to menus as part of the overall system security, and application implementation plan. See Chapter 2, “Security Subsystems,” for additional information.

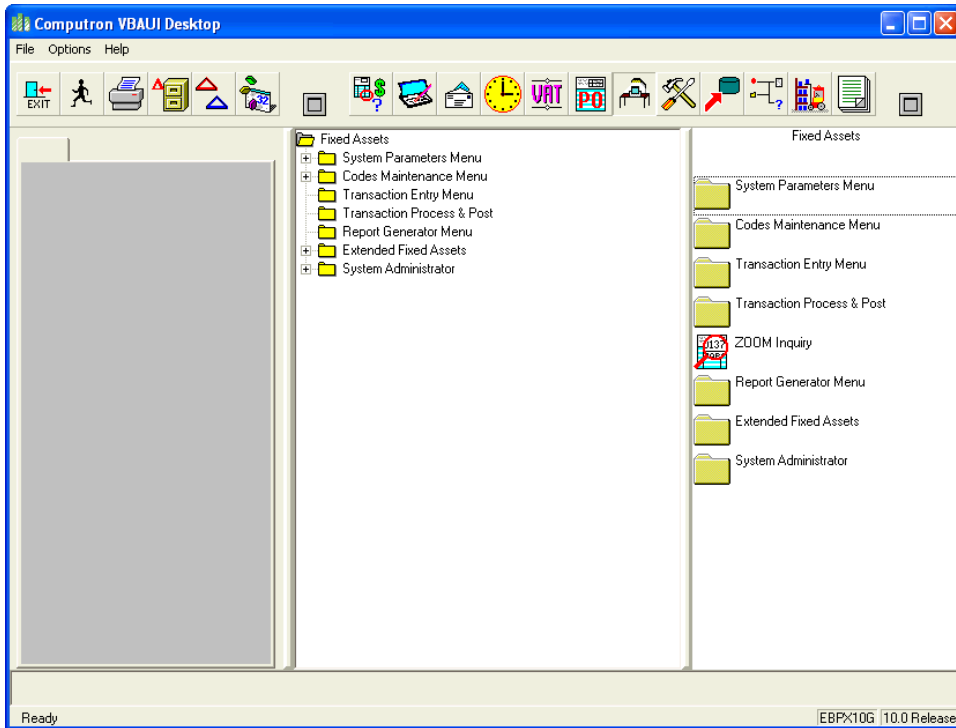
Before beginning this process, you should print a listing of the menus as they have been defined by running the Menu File Print function. See Section 1.5, “Menu File Print,” for additional information. This listing can be used as an audit trail, in case you inadvertently make unwanted changes to a menu and want to reset the options.



Menu File Print Screen (KS1)

Request

The standard Fixed Asset Master menu is displayed here:



Fixed Assets Master Menu

You are asked to:

- ◆ Establish security levels for each menu item (and/or submenus) in the Fixed Asset Master menu.
- ◆ Ensure that the tailored versions of several application reports (the reports have been tailored to include an additional field) are used by the system instead of the standard versions supplied by Computron.

Request — Establish Security

Establish the following security levels for each menu line item (and/or submenus) in the Fixed Asset Master menu:

Function Key	Menu Item	Security
PF1	System Parameters Menu	9
PF2	Code Maintenance Menu	2
PF3	Transaction Entry Menu	4

Function Key	Menu Item	Security
PF4	Transaction Process & Post	6
PF5	Zoom	1
PF6	Report Generator Menu	1
PF7	Extended Fixed Assets	7
PF8	System Administrator	9

☞ *Note:* Function key access for menus is only available in the character cell presentation; however the security parameters entered here will be followed by the associated icons in WEBdesk.

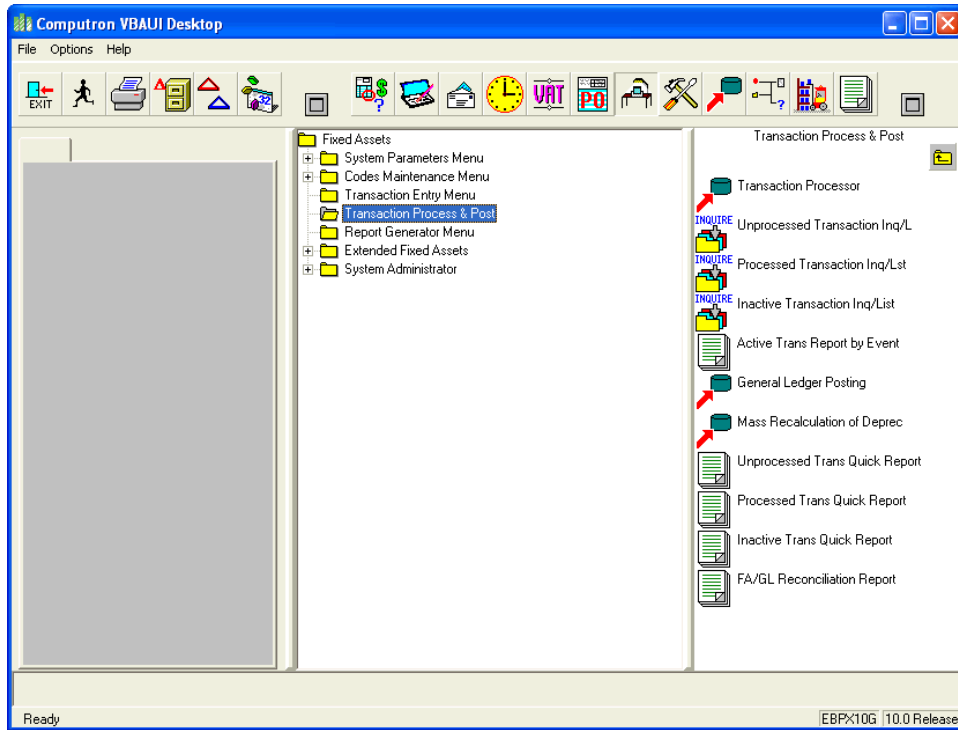
This means that the user must have an equal or higher security level (established for each user in Computron's User File Maintenance function) in order to select the option. Those menu selections to which the user does not have access are not displayed on the menu window.

☞ *Note:* There are two selections that require a security level of 9. At this site, one individual is responsible for system parameters (a Fixed Assets application expert), while another individual is responsible for system administrator functions (the system administrator who manages Fixed Assets); additional security options must be implemented to restrict access for each of these selections to the correct individuals.

Request — Accommodate Tailored Application Reports

Ensure that the tailored versions of several application reports (the reports have been tailored to include an additional field) are used by the system instead of the standard versions supplied by Computron.

This means that you need to modify the report ID used when the corresponding menu selection is made for the report. The standard versions of these reports are included on the Transaction Process & Post menu, as illustrated here:



Transaction Process & Post Menu

The tailored application reports are the Transaction Inquiry/List reports (Unprocessed, Processed, and Active).

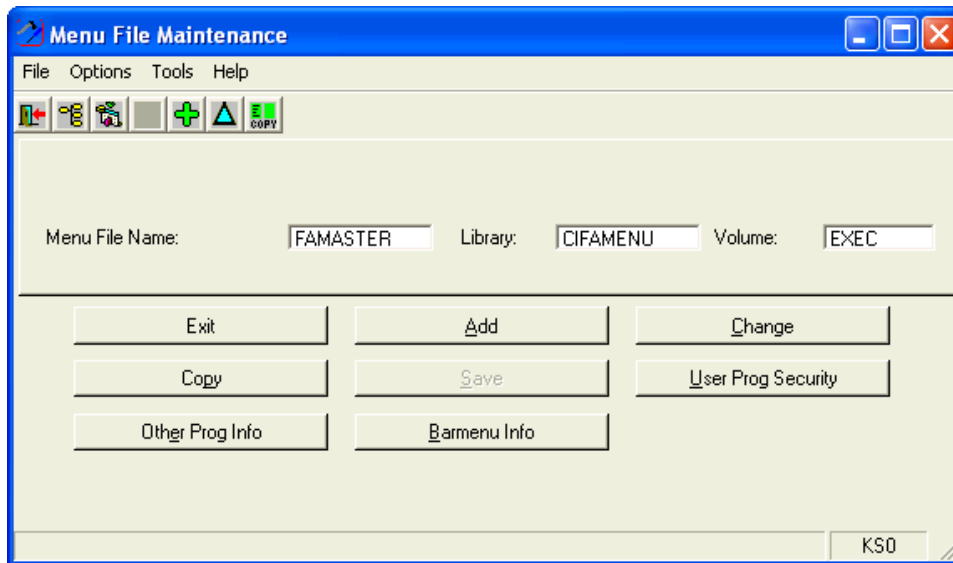
Computron Report ID	Tailored Report ID
TRAN%001	TRANC001
TRAN%002	TRANC002
TRAN%004	TRANC004

The tailored report IDs need to be substituted into the standard menu definitions for the Computron report IDs.

Solution

Solution — Establish Security

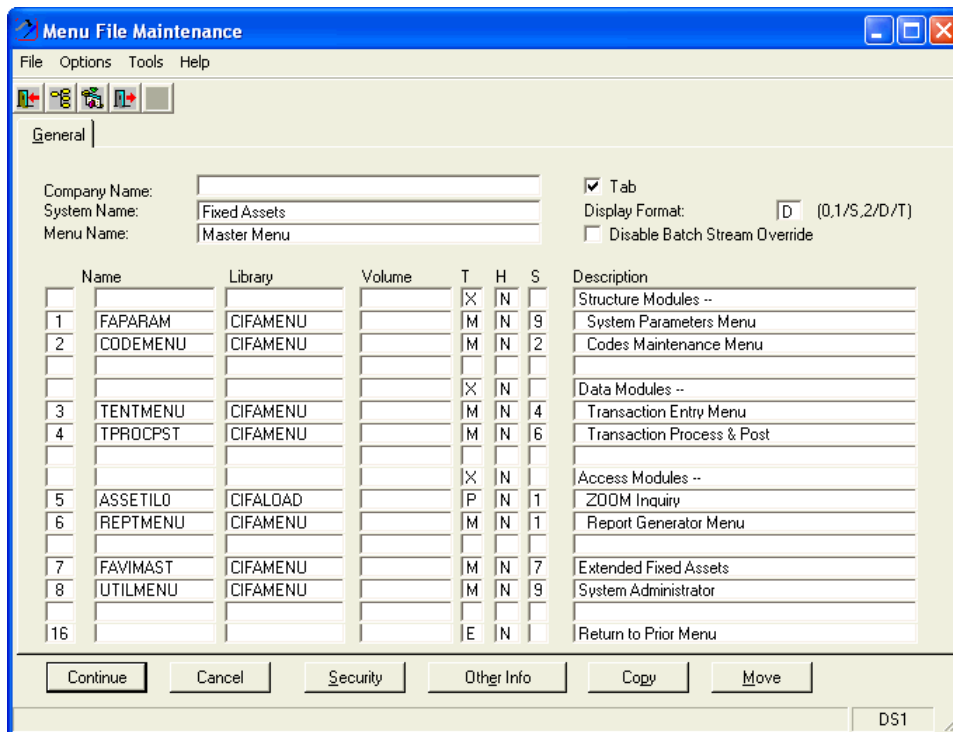
Access the Menu File Maintenance function and enter the Menu File Name/Library/Volume as shown here:



Menu File Maintenance Window (KS0) for FAMASTER

Note: The Save button (which saves menu changes) is not available or activated until after modifications have been made.

Click the Change button to modify an existing menu file, as shown here:

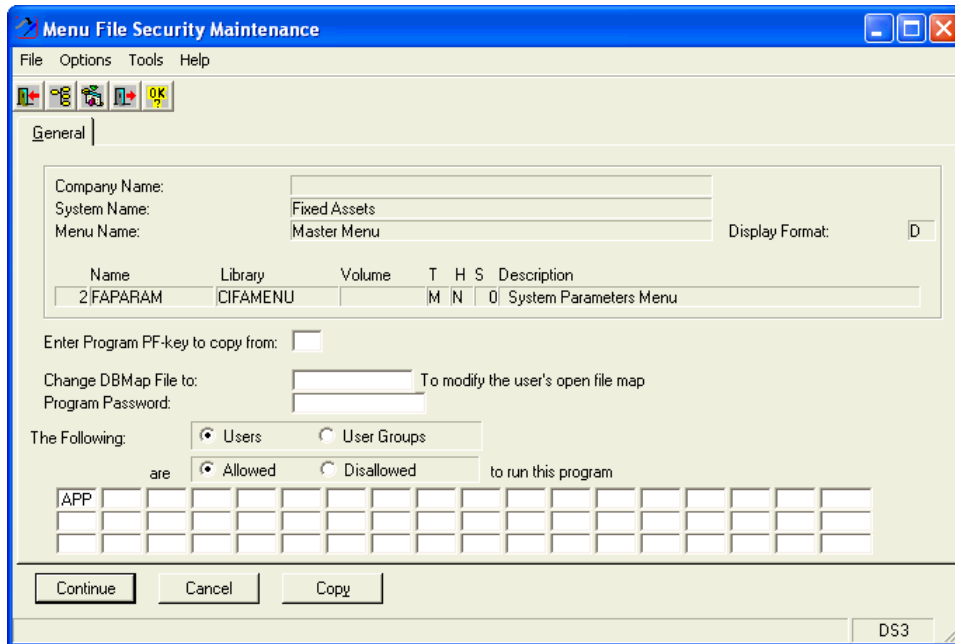


Menu File Maintenance Window (DS1) for FAMASTER

Enter the desired security levels in the S field. Specify particular security for the System Parameters menu and System Administrator menu selections as follows:

Selection	User ID
System Parameters Menu	APP
System Administrator Menu	SYA

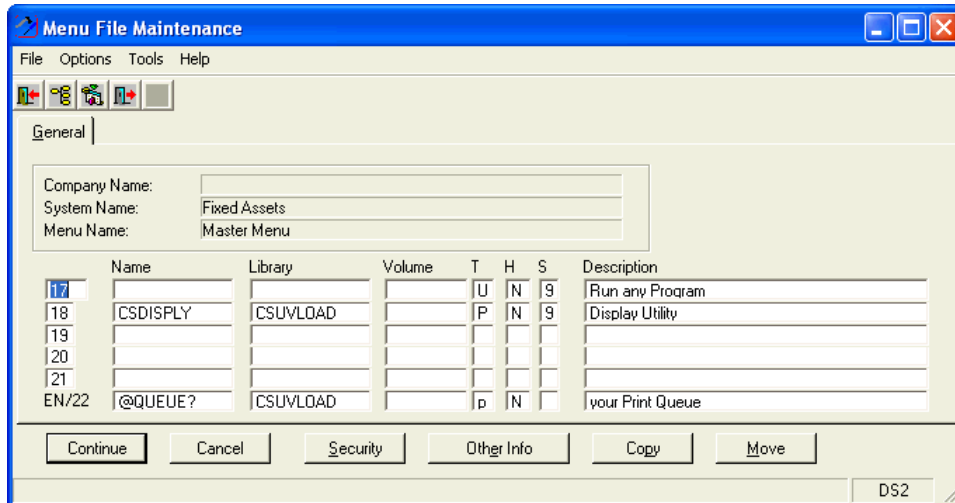
Only the user ID indicated above should be allowed access to the particular menu selection. In order to achieve this, position the cursor on the line for the System Parameters menu selection and click the Security button. The following DS3 window displays:



Menu File Maintenance Window (DS3) for FAMASTER

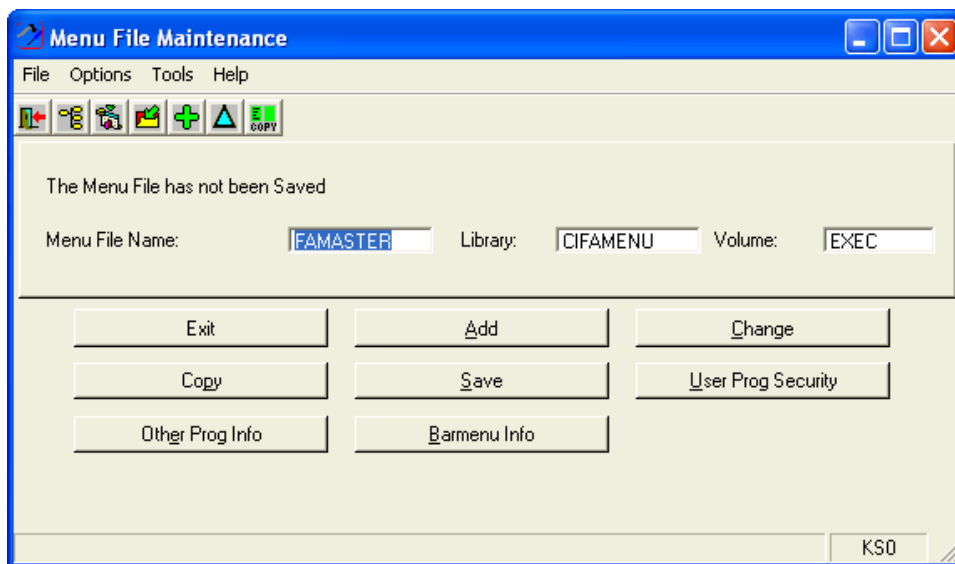
Click Continue to return to the original Menu File Definition window. Now modify the System Administrator menu selection in the same way.

You need make no additional changes to this menu, so click Continue to accept the second Menu File Definition Window (DS2) as it exists.



Menu File Maintenance Window (DS2) for FAMASTER

The Menu File Maintenance Menu Window (KS0) again displays, informing you that the Menu File has not been saved. Click the Save button.



Menu File Maintenance Window (KS0) for FAMASTER

Since the menu file already exists, an error message displays when you click Save. Click OK to overwrite (scratch) the existing file and save your modifications.

Solution — Accommodate Tailored Application Reports

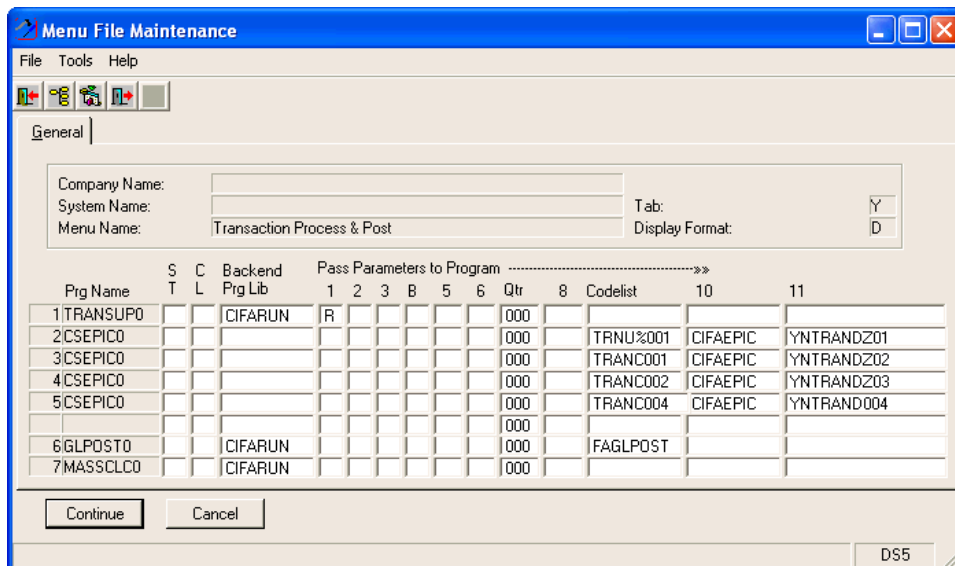
The changes to be made in this case are not security-related. The modifications can be accomplished in one of two ways:

- ◆ **Line-by-Line Basis** — If only one or two changes are needed on a menu, access the Menu File Maintenance Window (KS0) and click the Change button to modify the existing menu

file. Position the cursor on the appropriate line item and click the Other Program Info button.

- ◆ **Mass Change Basis** — If many changes are needed, access the Menu File Maintenance Window (KS0) and click the Other Prog Info button to display the Other Program Information Window (DS5) for the entire menu. This is the option used in the illustration that follows.

When the Other Program Information Window (DS5) displays, you can see the report IDs that are part of the standard Computron application system. The report ID for the list is in the Codelist column (e.g., TRNU%001); while the inquiry report ID is in the 11 column (e.g., YNTRANDZ01).



Prg Name	S	C	T	L	Backend Prg Lib	Pass Parameters to Program								Codelist	10	11	
						1	2	3	B	5	6	Qtr	8				
1TRANSUP0					CIFARUN	R							000				
2CSEPICO													000	TRNU%001	CIFAEPIC	YNTRANDZ01	
3CSEPICO													000	TRANC001	CIFAEPIC	YNTRANDZ02	
4CSEPICO													000	TRANC002	CIFAEPIC	YNTRANDZ03	
5CSEPICO													000	TRANC004	CIFAEPIC	YNTRAND004	
6GLPOST0					CIFARUN								000	FAGLPOST			
7MASSCLCO					CIFARUN								000				

Other Program Information Window (DS5) for TPROCPST

Enter the new report IDs in the Codelist field. The inquiry has not been tailored for this installation; therefore, do not modify the report ID in the 11 field.

Click Continue to return to the Menu File Maintenance window. Then, click Save, followed by the override button, to save the changes just completed.

☞ *Note:* You could save the tailored menus in a separate menu library, rather than in the standard Computron menu library (XXXXMENU); e.g., the standard Computron Fixed Asset system menus are stored in CIFAMENU. You could save tailored versions in CUFAMENU (CU is used in many sites to identify custom versions).

If this method is used, at the time an updated release of Fixed Assets is sent to your installation, the released version of the menu library does not write over your installation's tailored library.

☞ *Note:* If changes are made to the standard Computron menus, you need to modify your tailored versions to take advantage of these modifications. You should refer to the Computron consultant assigned to your site to determine the best way to approach saving menu changes for the site.

We recommend printing the Menu File Definition windows once again, keeping the Menu File Print file as an audit trail. Also test the operation of the menu system as it has been modified to ensure that the changes were made correctly.

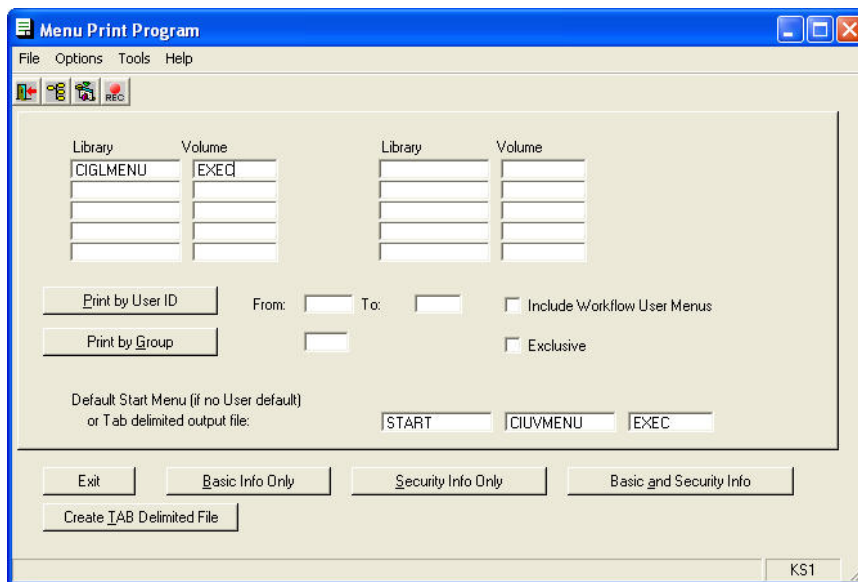
1.5 Menu File Print

Introduction to Menu File Print

This function is used to print a list of menu information as entered via the Menu File Maintenance function. You have the option to generate a report that contains basic menu information (i.e., information entered on the basic menu definition window), menu security information, or both.

Menu Print Program Window (KS1)

The following window displays when you select the Menu File Print function;



Menu File Print Window (KS1)

Library / Volume

These fields are used to select the Menu file(s) to print for any of the following options: Basic Info Only, Security Info Only, Basic and Security Info or Create TAB Delimited File. Enter up to ten menus (library / volume).

Print by User ID

Enter the range of user IDs for which you want to print menus. With this option, leave the Library and Volume fields blank and enter the name, library and volume of a valid menu in the Default Start Menu fields.

Include Workflow User Menus

This determines whether the Workflow menus for each individual user are included in the output. This field is used in conjunction with the Print by User ID option. Valid options are:

- Include menus created for individual users via the Workflow system.

- Do not include Workflow user menus.

Default:

Print by Group

Enter the group ID for which you want to print menus. With this option, leave the Library and Volume fields blank and enter the name, library and volume of a valid menu in the Default Start Menu fields.

Exclusive

This is used in conjunction with the Print by Group option. This prints the menus of those to which the group is expressly Allowed, based on the setting of the Allowed / Disallowed field in the Security information for the menu. Note that if a group is not allowed for a menu, sub-menus will not be accessed.

- Print only those menus that specify the selected Group is Allowed.
- Print all menus except those that specify the selected Group is Disallowed.

Default:

Default Start Menu (if no User default) or Tab delimited output file

Enter the location of the start menu for the Print by User ID option or Print by Group option or the location of the tab-delimited file for the Create TAB Delimited File option.

Default: START CIUVMENU IPLVOL

Options from the Menu Print Program Window (KS1)	
Option	Description
Print by User ID	Print the menus to which each user in a particular range has access.
Print by Group	Print the menus to which a group has access.
Exit	Return to the Universal Utilities menu.
Basic Info Only	Print basic menu information. This prints the menus and the Other Programs Information.
Security Info Only	Print security information. This prints the name of the function or menu and the security information associated with each.
Basic and Security Info	Print both the basic menu information and the security information. The output is two pages for each menu. One is the basic information; the next is the security information.

Options from the Menu Print Program Window (KS1)

Create Tab Delimited File

Create a tab-delimited file of the output data. To create this file, enter up to ten menus in the Library and Volume fields. Enter the location of the output file to be created into the field, Default Start Menu or Tab delimited output file.

This file can be imported directly into any application, such as a spreadsheet, where it would be useful to have the information in text format.

Menu File Print Report Description

The Basic Info Only, Security Info Only and Print by User ID formats are partially illustrated below. When the Basic and Security Info button is clicked, the report prints the basic information for a menu followed by the security information for the same menu. The format of the individual pages is the same as those illustrated here. The format for the Print by Group option is similar to the Print by User ID format.

Menu File Listing — Basic Info Only

```

*   Computron Software, LLC   ***** Thu Apr 26, 2007  2:31 PM *
*                               MENU REPORT - BASIC INFORMATION *
*                               ***** Page 175 *****
*
MENU FILE NAME UVUTIL  LIBRARY CIUVMENU  VOLUME EXEC
COMPANY NAME
SYSTEM NAME Universal Utilities
MENU NAME Master Menu

-----
NAME  LIBRARY  VOLUME T H S  DESCRIPTION  SC BACK END 1 2 3 B 5 6  QTR 8  CODELIST  10  11
-----
1  USERMNT3  CSUVLOAD  P N 0  User File Maintenance  000
2  CSEPIC0  CIRPLOAD  P N 0  User File Inquiry/List  000  USER#001 CSUVEPIC YNUSER#201
3  MENUMNT2  CSUVLOAD  P N 0  Menu File Maintenance  000
4  MENUPRT2  CSUVLOAD  P N 0  Menu File Print  000
5  DBMAPMNT  CSUVLOAD  P N 0  Data Base Map Maintenance  I 000
6  UVMAP  CIUVMENU  M N 0  MAP Maintenance  000
8  CSENTRY  CSUVLOAD  P N 9  Global Variable Maintenance  000  GLOBVAR CSUV
9  CSENTRY  CSUVLOAD  P N 0  Global Variable Inq/List  B 000  GLOBVAR CSUV
10 SCNIMNT  CSUVLOAD  P N 0  Soft Screen Maintenance  000
11 RESTART0  CSUVLOAD  P N 0  Restart Recovery  000 010
12 UVXUTIL  CIUVMENU  M N 0  System Utilities Menu  000
13 AUDUVPR0  CSUVRUN  P N 0  Print Audit File  CIRPLOAD
14 EPICMENU  CIUVMENU  M N 0  EPIC in UV  000
15  000
16  000
16  E N 0  Return to Prior Menu  000
17  U N 9  Run Any Program  000
18 CSDISPLY  CSUVLOAD  P N 9  Display  000
19  REGISTER .TXT
20  000
21  000
EN @QUEUE?  N 0  your Print Queue  000

```

Menu File Listing — Security Info Only

```

*   Computron Software, LLC   ***** Thu Apr 26, 2007  2:31 PM *
*
*                               MENU REPORT - SECURITY INFORMATION
*
*----- Page 176 *
*
*   MENU FILE NAME UVUTIL   LIBRARY CIUVMENU   VOLUME EXEC
*
*   COMPANY NAME
*   SYSTEM NAME Universal Utilities
*   MENU NAME Master Menu
*
*   NAME   PASSWORD A/D USERS (A)LLLOWED OR (D)ISALLOWED
*-----
1  USERMNT3      A
2  CSEPIC0
3  MENUMNT2
4  MENUPRT2
5  DBMAPMNT
6  UVMAP        A
8  CSENTRY
9  CSENTRY      D XX
10 SCNIMNT      A
11 RESTART0     A
12 UVXUTIL      A
13 AUDUVPRO
14 EPICMENU
      A
16
17
18 CSDISPLY
20
21
EN @QUEUE?

```

Menu File Listing –Print by User ID

```

*   Computron Software, LLC   ***** Thu Apr 26, 2007  2:31 PM *
*
*                               MENU SUMMARY FOR USER: PSE (pse   )   Pat Eckart
*
*----- Page 1 *
*
*   MENU FILE NAME START   LIBRARY CIUVMENU   VOLUME IPLVOL
*   SYSTEM NAME Financial Applications
*   MENU NAME 10.0 Release
*
*   NAME   LIBRARY   VOLUME T H S   DESCRIPTION   SC BACK END 1 2 3 B 5 6   QTR 8 CODELIST 10 11
*-----
01 GLMASTER CIGLMENU   M N 0 General Ledger   000
02 APMMASTER CIAPMENU   M N 0 Accounts Payable 000
03 ARMMASTER CIARMENU   M N 0 Accounts Receivable 000
04 TBMASTER CITBMENU   M N 0 Time Billing      000
05 VTMMASTER CIVTMENU   M N 0 VAT Module       000
06 POMAST    CIPOMENU   M N 0 Purchasing       000
07 FAMASTER CIPAMENU   M N 0 Fixed Assets     000
08 UVUTIL    CIUVMENU   M N 0 Universal Utilities 000
09 BSMMASTER CIUVMENU   M N 0 Job Processor     000
10 STARTWF   CIUVMENU   M N 0 Process Manager   000
11 IVMAST    CIIVMENU   M N 0 Inventory         000
12 EPICMENU  CIRPMENU   M N 0 Epic Report & Query Generator 000
16          EN N 0 Return to Prior Menu 000
EN          U N 9 Run any Program
EN CSDISPLY CSUVLOAD  P N 9 Display Utilities
EN @QUEUE?  p N 0 your print Queue

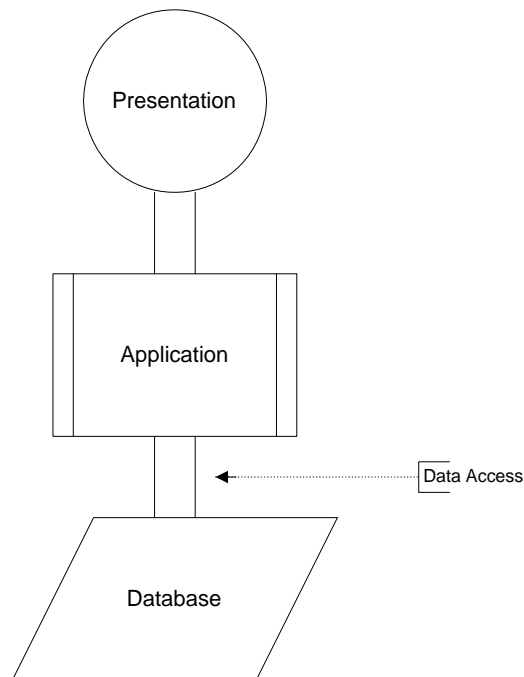
```

1.6 Database Map File Maintenance

Introduction to Data Base Map Maintenance

A simplified representation of Computron's application software can help illustrate the three-tiered architecture of the Computron system: presentation, application logic, and database.

Fig. 1.6-1
Computron's 3-Tier Architecture

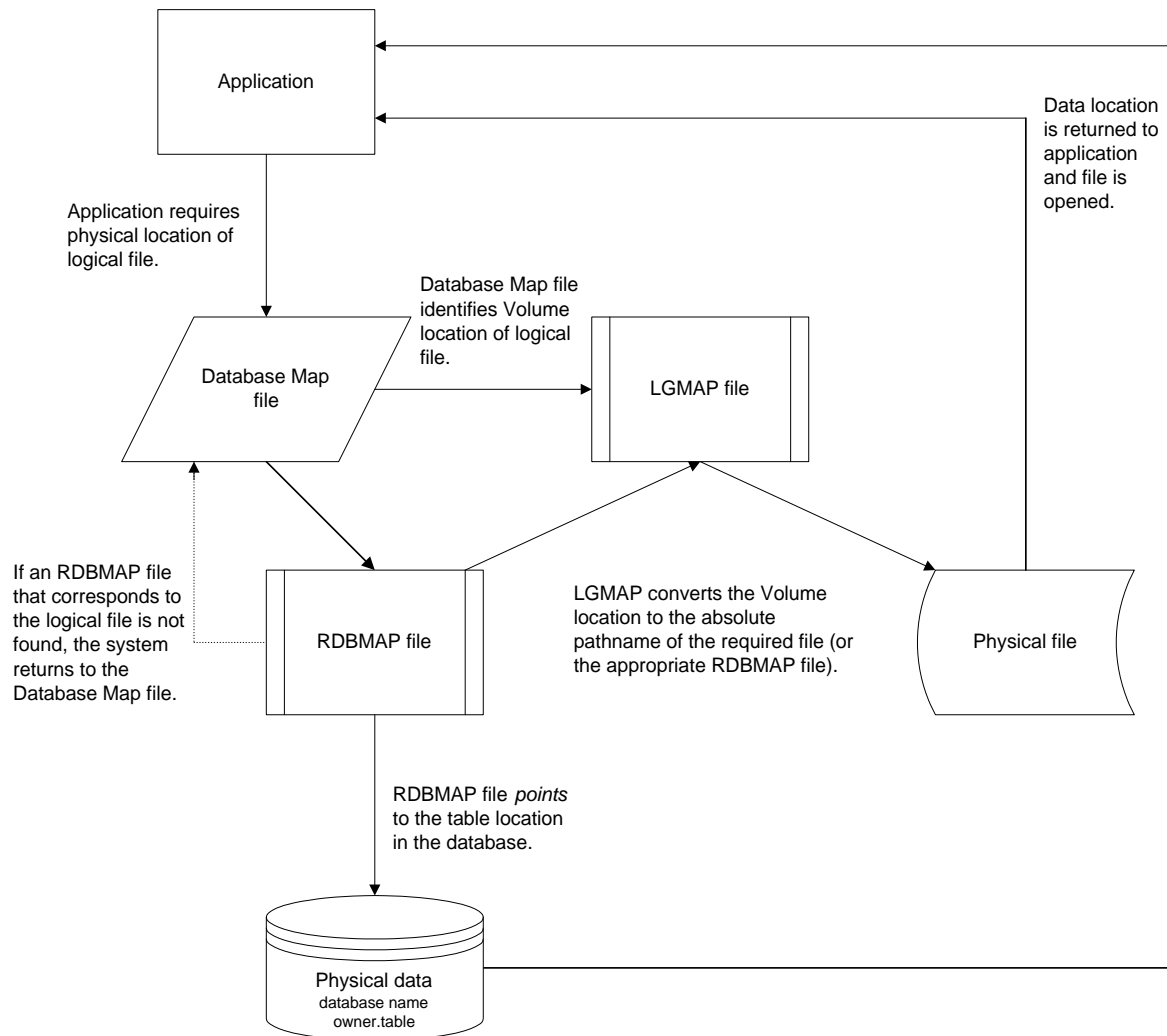


Computron applications locate data by reading the Database Map file in conjunction with the LGMAP file. An application program identifies required data by a logical name, which is comprised of a logical filename and a logical library. The Database Map file defines the physical location of logical files and libraries in the format: Volume, Library, File. The LGMAP file translates the Volume specification into information that is used to determine the physical location of the required data. In most cases, the LGMAP file expands designated volumes into the absolute pathname to a directory. Special entries, however, that represent the relational database name and server, are also interpreted by the LGMAP file.

Relational database implementations of Computron software introduce the RDBMAP file into the data access sequence. The RDBMAP file provides information regarding the tables and columns in which the required data resides. Since configuration and parameter information required by Computron applications is stored in parameter files, the Database Map file remains the primary link from the application to the database. An entry in the Database Map file identifies the location of the appropriate RDBMAP file that must be read in order to locate the required data.

The diagram below depicts the data access sequence. Each component of the database element (the database map, the LGMAP file, and the RDBMAP files) is described in greater detail in the following sections of this chapter.

Fig. 1.6-2

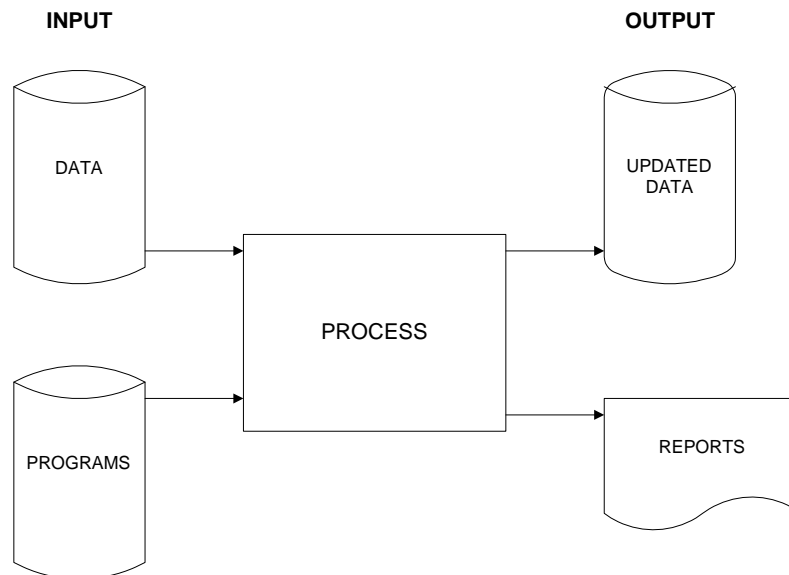


Data Access Flowchart

Database Map Maintenance

This function is used to maintain Database Map records for the Computron application. Before you can understand the structure and use of the Database Map file, you must first understand the concept of a database and how that concept is applied in Computron applications. A software system is commonly visualized as follows:

Fig. 1.6-3
Database Structures



This illustration depicts the components and operation of the software system. The components identified can be grouped into two basic types: programs and data.

- ◆ Programs — Programs comprise all of the code used to execute a program or procedure.
- ◆ Data — Data represents all of the facts stored in the system, as well as the information retrieved from the system (via reports). In the illustration, all data is conceptually stored in one place.

When you execute a program (usually initiated via a menu selection), the system assumes that the software is on the same volume as the menu program. Note that Computron uses the term *volume* to describe a six-character logical name that defines the location of the file or directory where the software resides. If the selected program does not reside in the same location as the menu from which the program is initiated (i.e., the program is not on the same volume as the menu), the menu file contains information to locate the program.

When a program requires access to data, it tries to find it by requesting the data by its logical name (logical library and logical filename). Note that Computron uses the term logical library to describe the application associated with the data request and the term logical filename to describe data recognized by

the application. The application request is processed by searching the database map for entries that correspond to the logical library and logical filename.

The Database Map File

The Database Map file is a single file that contains various entries for defaults and exceptions that direct the applications to Computron data. Thus, the database map serves as a road map to the data needed for processing. Depending on your database system, the Database Map file provides one of the following:

- ◆ Information to locate data files.
- ◆ Information to retrieve the data from a relational database, including the location of Computron RDBMAP files, the table owner of tables associated with the required data, the name of the relational database, and, where applicable, the name of the RDBMS server.

When you initially log onto the system, an entry in the User file indicates the default database map to be used. Whenever a Computron program is run, it looks at the user's database map to determine the location of the data it needs to access. The link between Computron programs and the database map is described in more detail in the sections that follow.

The LGMAP File

Due to the hierarchical nature of Unix file structures, the LGMAP file is used in conjunction with the database map. The LGMAP file contains the pathname for each volume called by a database map file. Essentially, LGMAP translates the logical volume names into absolute pathnames (or Native Location) that can be understood by the Unix system. For example, the LGMAP file expands relational database names and RDBMS server names beyond the length restrictions applied to those entries in the database map.

The LGMAP file resides in `/.../exec/_ctron_/wisp/config` (where `/...` represents the file system in which the Computron application software resides). Since all Computron system location information is stored in the LGMAP, entire Computron systems can be copied or modified by editing the LGMAP file (e.g., a production environment can be copied to a test environment).

Some sample LGMAP file entries may include:

```
WRKVOL      /.../var/worklibs
EXEC /.../exec
DATA /.../data
RDBMAP      /.../exec/rdbmap/release
SPLVOL      /.../var/spool
```

See Appendix C, "LGMAP File," for additional information.

Benefits of the Database Map

One of the benefits of the database map is that you access multiple databases with the same program without much effort. Under certain conditions, you can switch the database map simply by selecting the

Change Database Map function from the Computron desktop. This feature is particularly useful in situations such as system installation when you want to access both test and live data. In addition, a menu entry can automatically assign a database map for use at that and any subsequent menu/program levels, and then restore the original database map upon exit from that menu.

Another advantage of the database map is that the process of accessing multiple sets of data requires no corresponding programming changes. Typically, programs require data to be identified by a physical location. Therefore, to switch data without the benefit of a database map, you would have to edit the appropriate programs or scripts and change every reference to the original data.

To eliminate this extra step, Computron programs identify data by a logical file and library name only (e.g., the logical file `VOUCHER` in the logical library `CIAP`). Based on this logical name, the database map directs the program to the physical location of the required data. The location of the logical file called by a Computron program is defined by its actual file, actual library and volume entries in the database map. Subsequently, the `LGMAP` file converts the database map entries into location information that the data management system can interpret (e.g., the absolute pathname of the logical volume).

☞ *Note:* Relational implementations require `RDBMAP` files to be interpreted prior to retrieving data. The `LGMAP` file expands volumes to locate the appropriate `RDBMAP` file on the Unix system, and uses the `@RDBNAME` database map entry to translate logical library and volume entries into relational database names and `RDBMS` server names. The `RDBMAP` file supplies the table and column information that corresponds to the logical filename.

For detailed information on maintaining the `RDBMAP` file, refer to Appendix D, “`RDBMAP Maintenance`,” in this guide.

Examples

In certain `RDBMS` implementations, the Inventory and Purchase Order applications share the same Purchase Order text file. Therefore, the Unix pathname to the actual Purchase Order text file, which is represented in the database map as the logical file, `CIIVTEXT`, in the logical library, `CIIV`, is constructed as follows.

Logical Library	Actual		Logical File	Actual File
	Library	Volume		
CIIV	CIPOINT		CIIVTEXT	CIPOTEXT

When the Computron software is installed in a `/ctrn` file system, the `LGMAP` file entry for the volume name, `DATA`, appears as follows:

```
DATA /ctrn/data
```

Therefore, the absolute pathname for the `CIPOTEXT` file is returned as:

```
/ctrn/data/cipoint/cipotext
```

☞ *Note:* Since all actual file names are mapped to lower case, the case of the logical names for Volume/Library/File within a Computron application is irrelevant.

Sample Vision Database Entries

For each Computron application, seven basic entries are included in the database map. These database map entries are documented in each application system's *Technical Guide*.

Logical Library	Actual Library	Logical File	Description
XXXX	XXXXDATA		Data files for the application.
XXXX	XXXXSCNI	SCNIMAGE	Screen Image files for the application.
XXXXCTL	XXXXCTL		Data Description Lists (DDLs).
XXXXCTLI	XXXXCTLI		Program Data Description Lists.
XXXXEPIC	XXXXEPIC		EPIC Report Definition Lists.
XXXAUDT	XXXXAUDT		Audit Files for the application.
XXXXWORK	XXXXWORK		Temporary work files for the application.
XXXXDOCU	XXXXDOCU		Online Documentation files.

In each case, the characters XXXX symbolize the system prefix (e.g., CIAP = Accounts Payable, CIGL = General Ledger, CSUV = Universal, etc.).

☞ *Note:* Universal libraries are used with every Computron application; therefore database map entries must exist for both Universal files and the application specific entries. For example, entries required to utilize General Ledger include those files for both CIGL and CSUV.

When the relational database runtime is applied, the data file (XXXXDATA) entry is replaced by a line item that identifies the application RDBMAP tables and the owner of the corresponding tables to be accessed (&&XXXX). Additionally, entries are required to identify the relational database and where applicable, the RDBMS database server.

Sample Relational Database Entries

The relational database map included in a release of Computron software contains the following list of entries for each system:

Logical Library	Actual		Logical File	Description
	Library	Volume		
@RDBNAME	CTC	Servename		RDBMS database.
XXXXMAP		RDBMAP		Location of RDBMAP files.
XXXX	CTRONSYS	&&XXXX		Application RDBMAP files and owner of corresponding tables.
XXXX	XXXXPARM	EXEC		Application parameter files.
XXXX	XXXXWORK	WORKVOL		Application work files.
XXXX	XXXXSCNI	EXEC	SCNIMAGE	Screen Image files.
XXXXCTL	XXXXCTL			Data Description Lists.
XXXXCTLI	XXXXCTLI			Program Data Description Lists.
XXXXEPIC	XXXXEPIC			EPIC Report Definition files.

The @RDBNAME entry represents the database (or instance) of the database management system with which Computron's application software interfaces. Each database map can contain only one @RDBNAME entry. In this case, the database (instance) is called CTC. Installations where the relational database is accessed from a dedicated server must represent the server on the @RDBNAME database line item. In general, the Volume entry defines the server name and the Library entry defines the database (instance). The actual server name can be entered or an alias for the server name can be entered; however, alias server names must be defined in the LGMAP file.

For each application logical library, the location of the RDBMAP files and the owner of the tables described by the RDBMAP files must be entered. The Actual Library entry represents the table owner.

- ☞ *Note:* The application RDBMAP file entries must precede the application parameter file entries. Also a volume entry must accompany the application parameter file entry, even when the volume where the parameter files reside is the same volume where the database map resides.

Generic Database Map Entries

As an alternative to individual entries for each of the actual libraries listed earlier for their corresponding application logical library, the database map can include generic entries for the Screen Image, Data Description Lists, EPIC Report Definitions, RDBMAP files, Audit files and Online Documentation files. These entries are:

Logical Library	Actual Library	Actual Volume	Logical File	Description
&CTL				
&SCNI				
&EPIC				
&AUDT				
&MAP				
&DOCU				

To use the generic entries, all the similar actual libraries must exist on the same volume. For any actual libraries that are in a different location, an exception entry must be included in the database map. Any associated data libraries (CIXX) must be specified by individual entries for each application system (CIGL, CIAP, etc.).

Notes:

- ◆ The &CTL entry is used by the system to find the location of both the XXXXCTL and XXXXCTLI libraries. These libraries must always be located on the same volume.
- ◆ In most installations, the Screen Image Libraries are released as XXXXSCND. As part of the installation procedure, these libraries are renamed to XXXXSCNI so that subsequent releases of the XXXXSCND library do not automatically overwrite any customizations made by Computron's Visual Basic Forms Editor or Soft Technology tools.

If all the libraries specified via generic entries exist on the same volume as the database map, the generic entries can also be excluded, with the exception of the Screen Image entry (&SCNI). Thus, in its simplest form, the database map would contain entries for the location of the user data (XXXXDATA or &&XXXX and XXXXPARM) only.

- ☞ *Note:* Where an explicit entry for a Logical Library is included in the database map, it always takes precedence over the generic entries in that database map.

Special Database Map Entries

Depending upon the environment, special entries are included in the database map for one of the following reasons.

Multiple Applications Are Sharing Data

For example, the General Ledger and Accounts Payable systems both access the logical file, AUTOBAL. The AUTOBAL data is stored with the General Ledger data; therefore, when the Accounts Payable application is looking for the AUTOBAL logical file in the CIAP logical library, the database map must direct it to the General Ledger data. The following entry tells the system to locate the data in the appropriate General Ledger table, which in this example is owned by CTRONSYS.

Logical Library	Actual		Logical File	Actual File
	Library	Volume		
CIAP	CTRONSYS	&&CIGL	AUTOBAL	

Text File Maps for Sybase and Informix Databases

The Sybase and Informix RDBMS implementations require separate RDBMAPS for the text files associated with Computron applications. Therefore, the database map used with Unix implementations must have separate text map entries for each available application. For example, if you are running the General Ledger, Purchasing, Inventory, Time Billing (TEAM), and Fixed Assets applications, there must be corresponding actual library entries for CIGLTEXT, CIPOTEXT, CIIVTEXT (points to CIPOTEXT), CITBTEXT and CIFATEXT in the database map.

Work File Locations

Many functions create work files during processing. Controlling the location of work files that are created by certain applications can optimize efficiency. As an example, the following entry can be added to the database map for the Accounts Payable system:

Logical Library	Actual		Logical File	Actual File
	Library	Volume		
CIAP	&WORK	WRKVOL	APREGPS1	&UT#APS1

Note: This example also includes a variable Actual File entry, which is explained in this following section.

Duplication of Filenames

The use of certain features requires data that is not included in the same logical library as the standard data (e.g., work files that are placed on a different volume). The actual filenames may not be known prior to runtime, due to the fact that variables are used for generating the actual filenames. This is typically

used for report and work files, to avoid duplication of filenames when multiple users are running simultaneously.

The filenames are generated (and entered in the database map) using one or more of the following variables:

&UT	The User Task number (3 positions).
&UT#	The User Task number (4 positions).
&ID	The User ID (3 positions).
&DATE	The System data (6 positions or 5 positions if concatenated with &ID or &*UT).
&TIME	The system time (8 positions).

For example: The actual filename for the General Ledger EDETRANS file can be entered in the database map as follows:

Logical Library	Actual Library	Actual Volume	Logical File	Actual File
CIGL	&WORK	WRKVOL	EDETRANS	&UT#EDE

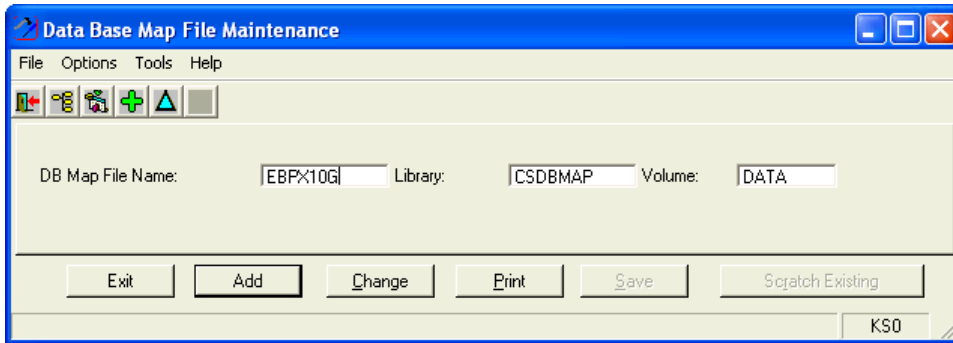
If the current User Task number is 3623, the actual filename created is 3623ede, which appears in the Native Location column.

Sample Actual File Entries

Logical Library	Actual Library	Actual Volume	Logical File	Actual File
CIAP	&WORK	WRKVOL	APREGPS1	&UT#APS1
CIAP	&WORK	WRKVOL	PAYVOU	&UT#PAY
CIGL	CIGLWORK		EDETRANS	&UT#EDE
CIGL	CIGLWORK		GLTRNFMT	&UT#FMT

Database Map Maintenance Window (KS0)

The Database Map File Maintenance (KS0) key window allows you to access the Database Map File Maintenance options. Upon opening the function, by default your current database map appears in the File Name/Library/Volume fields. To proceed to a particular maintenance option you can either accept the current database map or enter the name and location of another database map to maintain.



Database Map File Selection Window (KS0)

DB Map File Name/Library/Volume

This is the name and location of the Database Map file to maintain.

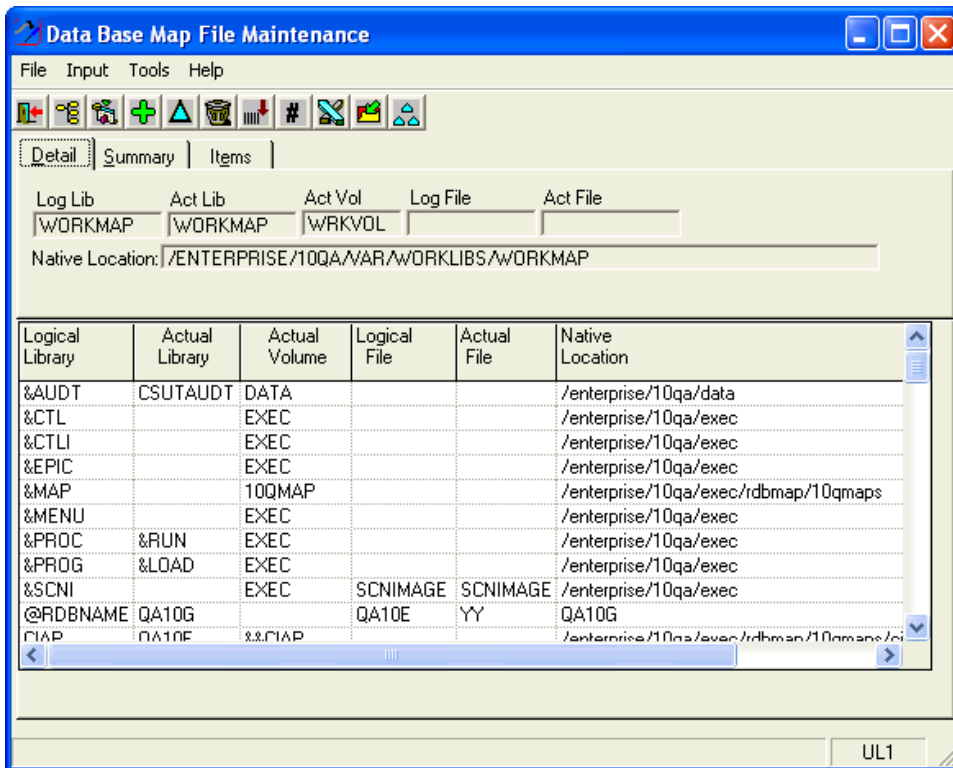
Table 1.6-1 Options for the Database Map File Maintenance Window (KS0)	
Button	Function
Exit	Exit the Database Map File Maintenance function.
Add	Add a database map. Proceed to the Database Map File Maintenance Window (DS1) where you can begin filling in the highlighted fields on the window.
Change	Modify an existing database map. Proceed to the Database Map File Maintenance Window (DS1) where you can make the changes to the logical and physical libraries that constitute the database map.
Print	Print the Database Map File.
Save	Save any of the changes or additions made during this session. If you exit this function before clicking the Save button, any database map entries added or changed are not saved.

Table 1.6-1 Options for the Database Map File Maintenance Window (KS0)	
Button	Function
Scratch Existing	<p>If you are saving changes made to an existing database map file, the system displays a message below the Data Base Map File Name/Library/Volume fields as a reminder:</p> <pre>The Database Map File Has Not Been Saved</pre> <p>When you click the Save button on the Database Map File Maintenance Window (KS0), a message displays in the status bar:</p> <pre>File already exists press PF17 (Scratch Existing) to scratch it.</pre> <p>To overwrite the existing file you must click the Scratch Existing button.</p>

☞ *Note:* The Save and Scratch Existing buttons are not available on the Database Map File Selection Window (KS0) until after a database map has been opened and modified.


Data Base Map File Maintenance Window (UL1)

Enter the appropriate information on the key Database Map File Maintenance Window (KS0) and then click the Change or Add button to display the Database Map File Maintenance Window (UL1).



Database Map File Maintenance Window (UL1)

This window allows you to maintain the names and locations of the actual (physical) libraries that are a part of the database map. Since this is a UL type (Universal Line Method) window, you can maintain multiple line items simultaneously. The libraries that constitute the Database Map file are displayed as line items, each of which can be operated on independently.

To edit an existing line item, first select it and then click the Change toolbar button . Note that the window switches into Change Mode and that the segments that constitute the line item can now be modified via the fields described here:

Log Lib

Enter the logical name of the library as it is referenced in the Computron program. Normally, this name corresponds to a system prefix. For example, CIAP for Accounts Payable, CIGL for General Ledger or WFAR for TAC.

Act Lib

Enter the name of the corresponding directory as it is identified on the disk.

Act Vol

Enter the logical name that defines the physical location of the library.

Default: If left blank, the volume defaults to the volume where the database map resides. If the parameter file entry does contain a blank volume entry, however, the system may re-sort the database map entries causing the parameter files to be accessed ahead of the relational tables.

- ☞ *Note:* If the remainder of the line item is left blank, the program accesses the default library indicated above and searches for an actual file with the same name as the logical file.

The next two fields, Log File and Act File, are used only when one or both of the following two exceptions occur:

- ◆ Exception #1 – The actual data specification differs from the logical name in the program.
- ◆ Exception #2 – The data or RDBMAP file is stored in a library or on a volume that differs from the default actual library or volume specification. For example, suppose the Posting Accounts data is stored in a table owned by a different RDBMS user than the standard owner established for General Ledger. You would, therefore, establish a separate entry for the Posting Accounts file to specify the different table owner.

Log File

Enter data in this field when either of the above exceptions occurs.

In the case of Exception #1, this field contains the name of the logical file whose corresponding physical file has a different name.

In the case of Exception #2, this field contains the name of the logical file with different library or volume entries from the standard.

For an example of possible field entries, refer to the examples that follow.

Act File


Enter data in this field only when Exception #1 occurs. It contains the name of the actual file whose name differs from the logical file name indicated in the program.

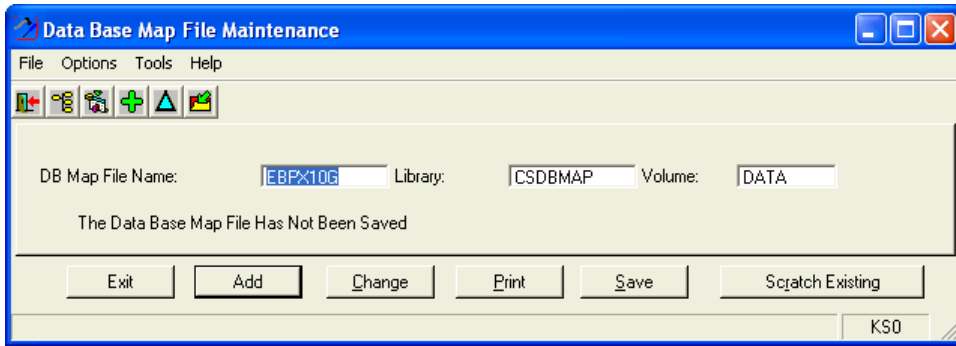
Native Location

This display-only field shows the absolute pathname for a selected line item, based on the current LGMAP file.

Saving Your Database Map Changes

When you're finished adding, modifying, or deleting any line items, follow these steps to save your changes:

1. Depending on what maintenance mode you're in, click the Add, Edit, or Delete button at the bottom of the data entry area of the window. The UL1 window returns to non-edit mode and your changes appear in the list box.
2. To process your changes, click the Save toolbar button , which returns you to the KS0 window.




Database Map File Maintenance Window (KS0)

Since you are still in edit mode, the KS0 window informs you that the database map file hasn't been saved, and the Save and Scratch Existing buttons become available.

3. To update the file, click the Save button. A status bar message informs you that you must scratch the existing database map file.
4. Click the Scratch Existing button to overwrite the file with your changes. Note that the Save and Scratch existing buttons are now no longer available.
5. Exit the Database Map Maintenance function and then log off the Computron desktop.
6. Reopen the Computron desktop to use the database map with your changes.

Specifying Additional Database Parameters

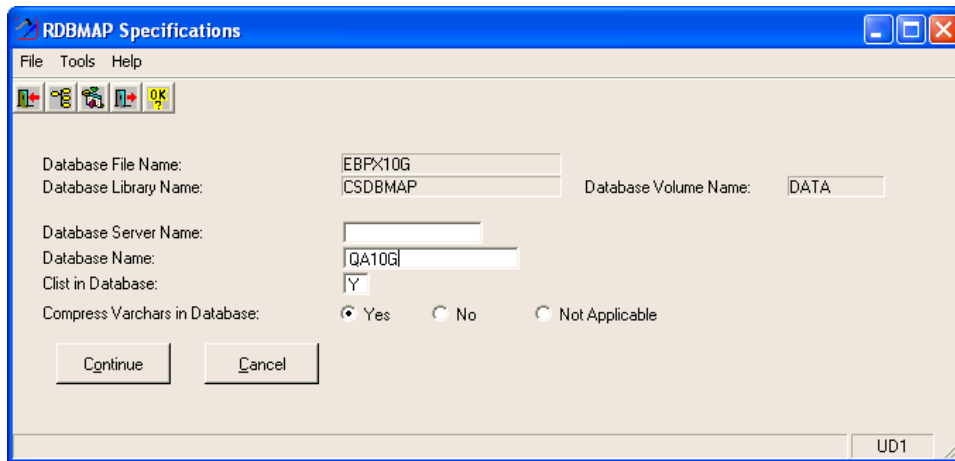
For RDBMS implementations, some additional database map parameters can be established on the RDBMAP Specifications Window (UD1). For instance, some implementations may need to update older versions of their database maps in order to access the database tables associated with code list (CLISTCTL) RDBMAPS. Moreover, Oracle and Informix implementations may want to maximize their disk space by eliminating trailing spaces (VARCHARS) in the database.

 *Warning! These settings should only be modified under the direction of Computron support personnel, as an improperly configured database may render the Computron system inoperable.*

The RDBMAP Specifications Window (UD1)

To access the RDBMAP Specifications Window (UD1) window from the Database Map Maintenance Window (UL1 window), follow these steps:

1. From the UL1 window, highlight the @RDBNAME line item, and then click the Change toolbar button.
2. With the @RDBNAME entry in edit mode, click the Edit button. The RDBMAP Specifications Window (UD1) displays.



RDBMAP Specifications Window (UD1)

3. To update your database map, enter and/or accept the appropriate field values:

Database Server Name

This is the name of the database server where the database resides.

Database Name

This is the name of the database where the Computron data resides.

Note: The Database Server Name and Database Name entries for the Database Map file may have alias names in the LGMAP file. For more information, refer to “The LGMAP File,” discussion earlier in this section.

Clist in Database

Normally defaults to Y. Older versions of Computron Software must enter a Y in this box to update the database map so that it can access the database tables associated with code list (CLISTCTL) RDBMAPS.


Compress Varchars in Database

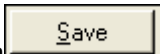
Normally defaults to Yes. Older versions of Computron Software must select the Yes option to remove trailing spaces (VARCHARS) are used in the database. Valid options are:

Yes – Remove trailing spaces in the database.

No – Do not remove trailing spaces in the database.

Not Applicable – This option does not apply to your database.

4. Click Continue to return to the UL1 window.
5. To save and process your changes, click the Save icon , which returns you to the KS0 window.

6. To update the database map file, click the Save button . A status bar message informs you that you must scratch the existing database map file.

7. Click the Scratch Existing button to overwrite the file with your changes. (Note that the Save and Scratch existing buttons are now no longer available.)
8. Exit the Database Map Maintenance function and then log off the Computron desktop.
9. Reopen the Computron desktop to access the database map with your changes.

Database Map Program Logic

In a Computron program, data is identified by the logical library where it resides and its logical name (e.g., CIGL, POSTACCT). To obtain the physical location of the data using these identifiers, the program does the following:

1. Reads the database map (as identified in the User file or on the Change Database Map window). If the library or volume is blank, the program uses the input library or volume from usage constants: The data volume name and a list of default logical libraries and logical files that give the actual physical library, file, and volume.
2. The program searches the database map entry for an exception entry of logical library and logical file. If it finds one, then it translates the file, library, and volume to the absolute pathname of the file via the LGMAP file and returns the physical location to the program to be used for opening the file.
3. If the explicit logical library and logical file entry are not found, then the program searches the database map entries to find the default logical library with a blank file entry. Relational database implementations contain two such entries: the first default logical library entry invokes the RDBMS and directs the program to a table for the data, while the second default logical library designates the location of the parameter files.

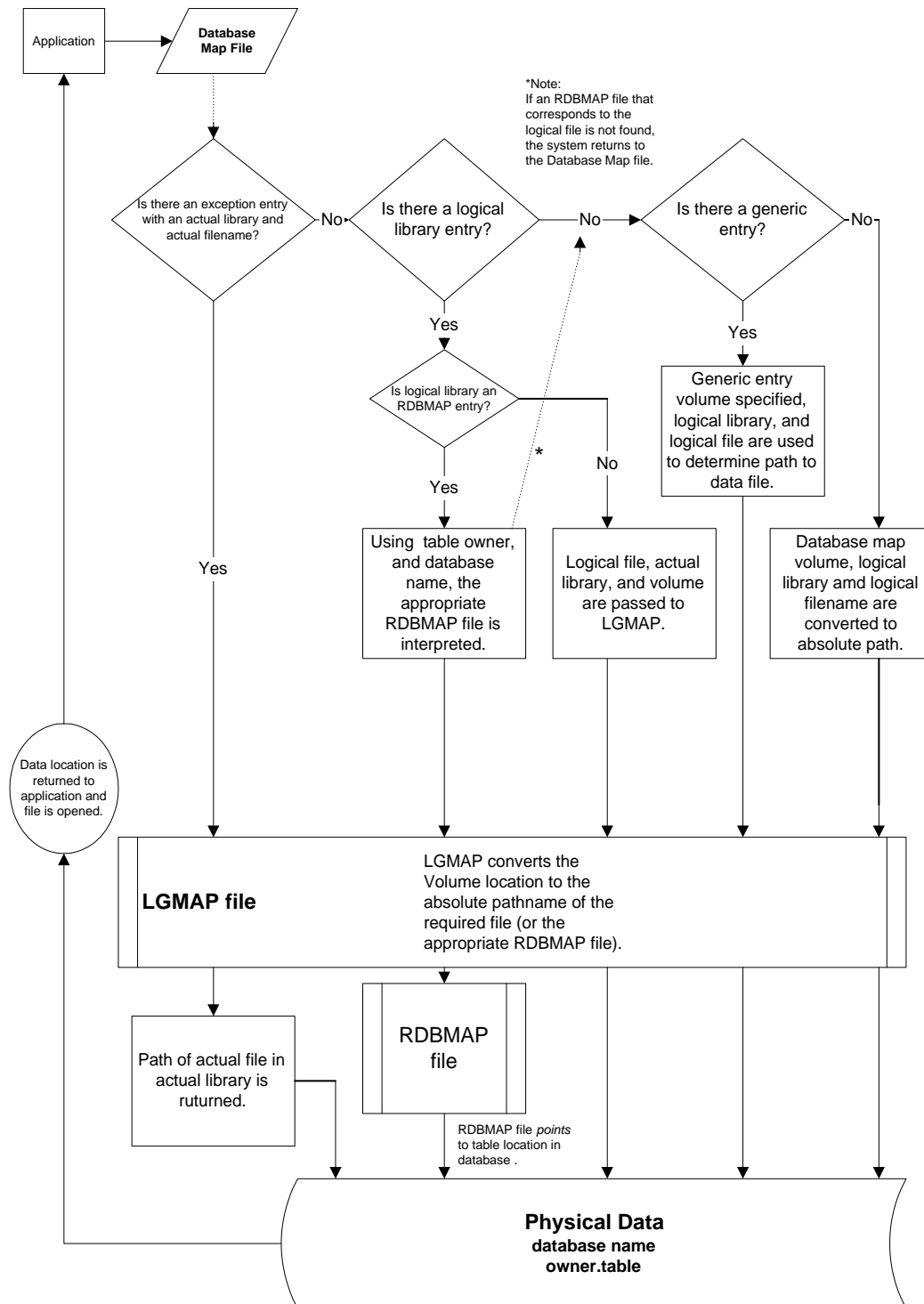
The program always searches the logical library encountered first in the database map for the required data. Thus, the program first interprets the RDBMAP file to determine the table location where the required data is stored. If found, the table information (table name, table owner, column, and row) is returned to the program and the data is retrieved. If the required data is not contained in a table, the program returns to the database map and continues searching for a default library.

Where such an entry exists (usually the location of the application parameter files is denoted via a logical library line item), the logical filename is assumed to be the actual filename. The actual library and volume are translated via the LGMAP and the absolute pathname of the required file is returned to the program. If the logical file is SCNIMAGE, however, the program utilizes Step 5.

4. If the program does not find an entry for the default library, then it searches for the corresponding generic entry (&CTL, &CTLI, &EPIC, &SCNI, or &SCND with the logical file SCNIMAGE) and converts the logical library, logical file and the actual volume to the actual pathname of the required file.
5. If the library is not found for a generic entry or a generic entry is not found, then the logical library, logical file and the database map volume are assumed to be the actual library, file and volume. The absolute pathname of the file is determined and returned to the program.

The following illustration depicts this flow.

Fig. 1.6-4
Computron's Data Access Flow



Examples

The examples shown here demonstrate how the Logical File and Actual File fields can be utilized.

Example #1 Computron relational database implementations may store some temporary data in parameter files rather than in tables. Also, if auditing is enabled, audit information is written to sequential log files. Those records stored as parameter files, or as log files, are entered in the database map using the logical library, actual library, and logical file entries as follows:

Logical Library	Actual		Logical File	Actual File
	Library	Volume		
CIGL	CIGLPARM		EDETRANS	
&AUDT	CSUVAUDT	DATA		

Example #2 There may be situations where data for one application is needed from a table that corresponds to a different Computron system. This situation occurs when both Computron's Accounts Payable and General Ledger systems are installed. The General Ledger logical file, AUTOBAL, is accessed via the Accounts Payable Voucher Entry function. In order to access the appropriate data, the following database map entries point the application to the General Ledger RDBMAP files to locate the table information for the AUTOBAL data.

Logical Library	Actual		Logical File	Actual File
	Library	Volume		
CIAP	CTRONSYS	&&CIGL	AUTOBAL	

Example #3 Many functions create work files during processing. Controlling the location of work files that are created by certain applications can optimize efficiency. The following entry can be added to the database map for the Accounts Payable system:

Logical Library	Actual		Logical File	Actual File
	Library	Volume		
CIAP	&WORK	WRKVOL	APREGPS1	&UT#APS1

- ☞ *Note:* This example also includes a variable Actual File entry as explained earlier in this section. These entries are typically used for report and work files, to avoid duplication of filenames when multiple users are running simultaneously.

1.7 Global Variable Maintenance

Introduction to Global Variable Maintenance


Global variables are used in Computron systems to keep track of sequential numbers assigned as part of record generation. For example, a global variable is used to control transaction numbers assigned to transactions within each journal in the General Ledger system. In addition to storing the last number used, each system maintains the highest and lowest values that can be assigned to a record.

Thus, if the following global variables are established,

Start Range	1
Ending Range	999
Last Number Used	998

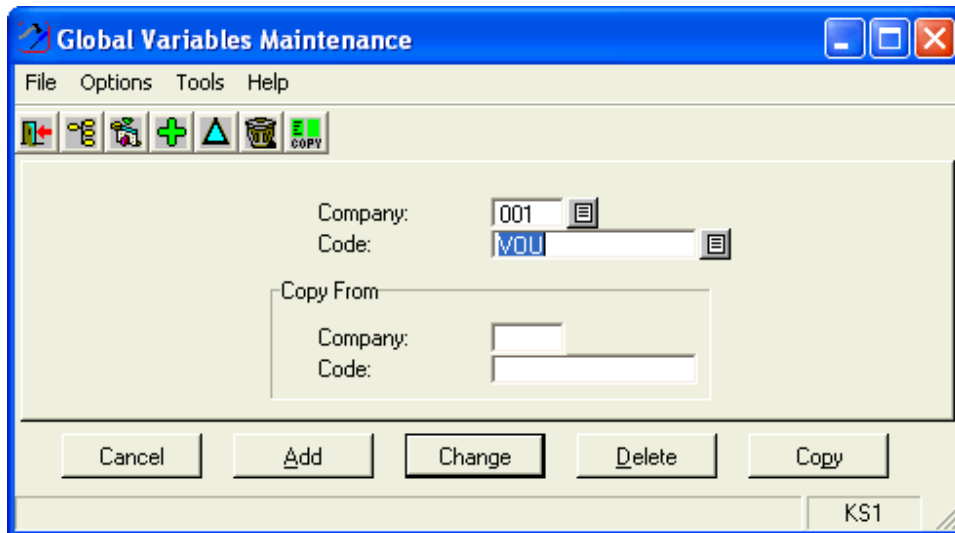
after the number 999 is used, the system assigns the number 1 to the next record.

Global Variable Maintenance is used to maintain global variables that have been previously established via system functions. Normally, variables are set by the application programs that use the variables. In addition, many Computron applications are installed with predetermined global variable settings. It is not normally necessary to maintain global variables. However, Global Variable Maintenance is provided in the event that some system failure (e.g., hardware failure) corrupts global variable settings.

 *Warning: This function should only be used with guidance from Computron's support staff. In addition, it is important to note that most Computron applications are designed not to reassign a number that has been previously assigned. Therefore, it is generally not necessary to change these ranges.*

Global Variables Maintenance Window (KS1)

The KS1 window displays when you launch the function.



Global Variables Maintenance Window (KS1)

Company (Q-Mark)

Enter the company number for which you are establishing/maintaining the global variable code.

Code (Q-Mark)

Enter the global variable code to establish/maintain. Valid entries in this field differ according to the application you are running.

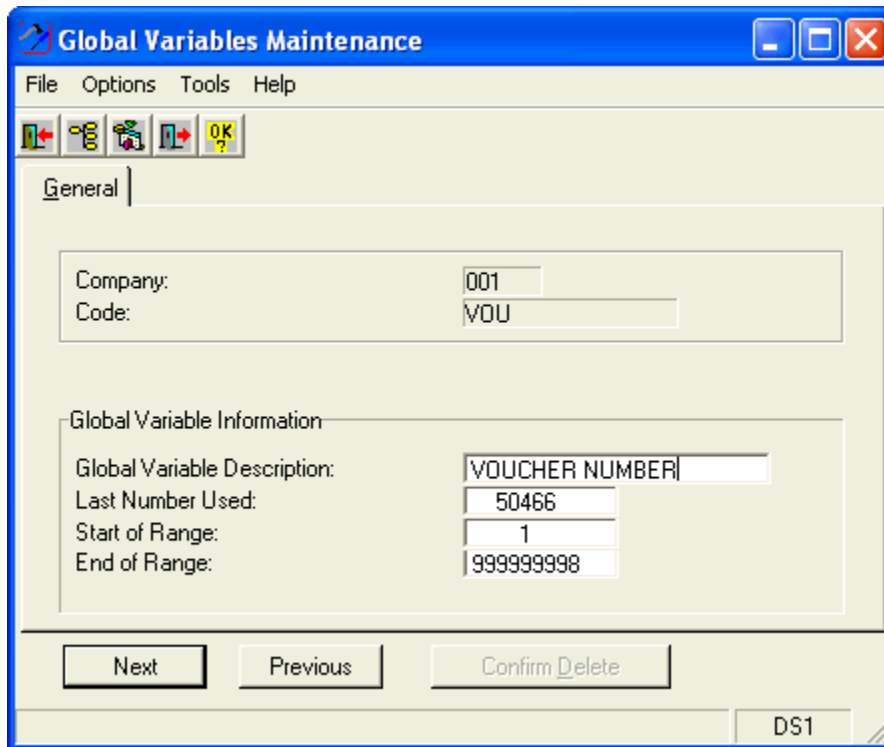
Copy From Company Code:

Enter the company number and global variable code from which a global variable is being copied. These fields are used only in conjunction with the Copy button.

Choose a maintenance option (Add, Change, Delete, or Copy) by clicking the appropriate button. The DS1 window displays.

Global Variables Maintenance Window (DS1)

The DS1 window displays from the KS1 window.



Global Variables Maintenance Window (DS1)

Global Variable Description

Enter a fifteen position alphanumeric description of the global variable. If the variable is created dynamically by the system, this description is the user ID of the person who was responsible for the creation.

Last Number Used

This is the last transaction number used. This number is incremented by one each time a transaction using this global variable is created.

Start of Range

A user-defined number to indicate the lowest number the system assigns to a transaction. The system-default value is 1.

Example: If Start of Range is four, then the first transaction is assigned the number four.

End of Range

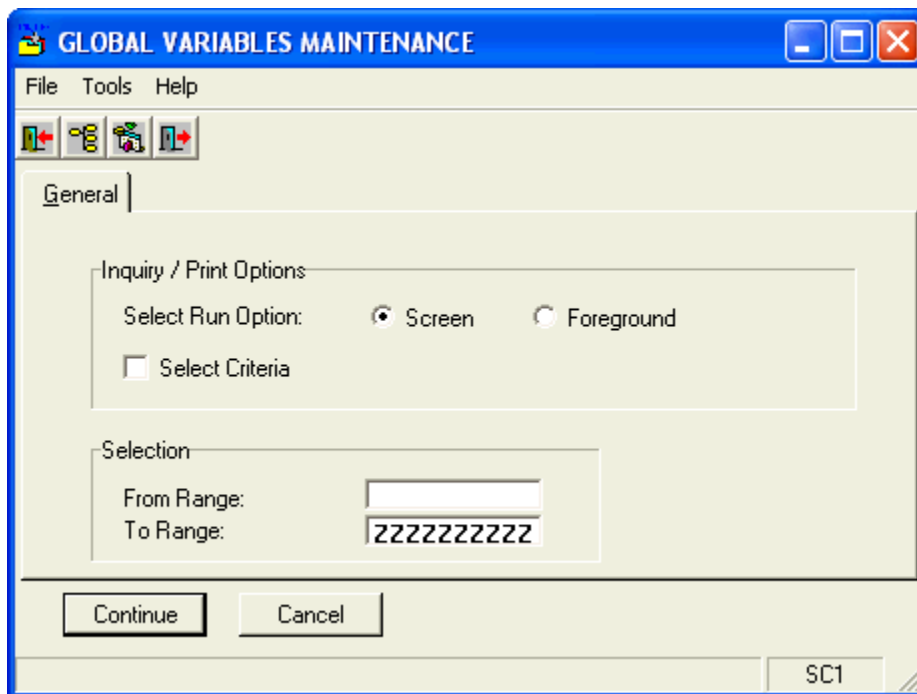
Maximum system-generated number assigned to a transaction using this global variable. If this number is reached, the system assigns the Start of Range number to the next transaction. In the example shown on the above window, once you enter the transaction number 999,999,998, the system assigns the next transaction number 1 and continues incrementing.

1.8 Global Variable Inquiry/List

Introduction to Global Variable Inquiry/List

This function is used to display a listing of the Global Variable codes established via the Global Variable Maintenance function. This data can either be displayed online or spooled to a print file.

Global Variable Maintenance Inquiry/List Window (SC1)



Global Variable Maintenance Inquiry/List Window (SC1)

Select Run Option

This is the processing option that produces online output or a printed report. Valid options are:

Screen – Enables you to review a list of the data online, as well as to display detail for each item on the list.

Foreground – Produces a print file, processing the request in foreground. Use of this option restricts the use of this window for other processing.

Default: Foreground.

Select Criteria

This determines whether to use a query window to specify additional selection criteria. Valid options are:

Displays the Select Query Window (SC99), which allows you to further define the selection criteria.

Do not allow modification of the selection criteria.

Default:

From Range

Enter the beginning of the range of codes to include in the inquiry/list.

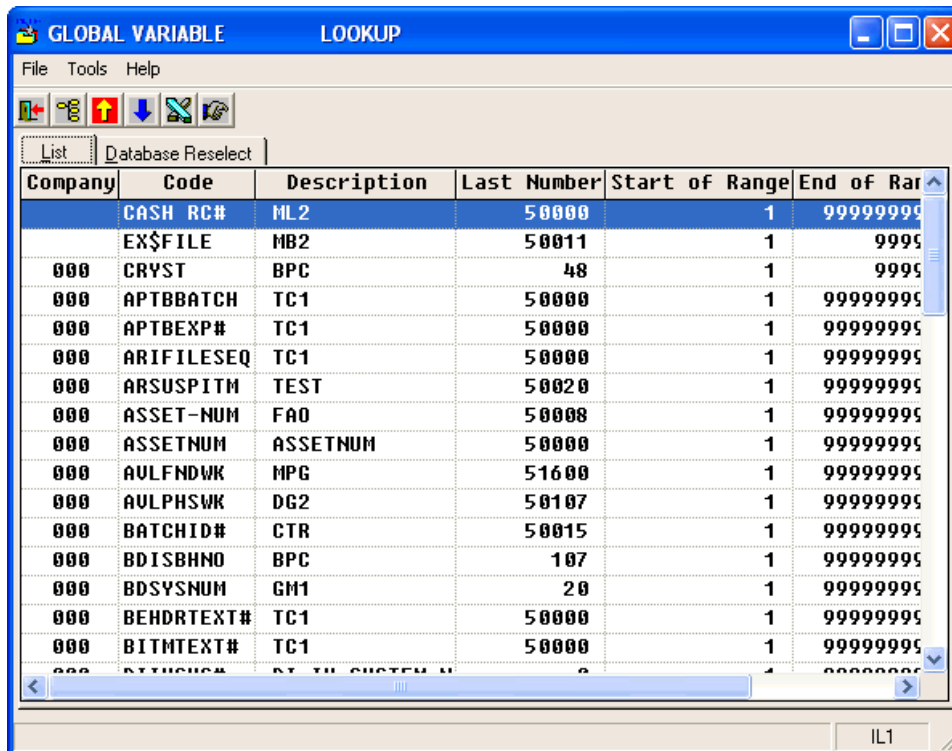
To Range

Enter the end of the range of codes to include in the inquiry/list.

Global Variables Report Options

Screen Option

When the Screen option is selected, an inquiry window displays:



The screenshot shows a window titled "GLOBAL VARIABLE LOOKUP" with a menu bar (File, Tools, Help) and a toolbar. Below the toolbar are buttons for "List" and "Database Reselect". The main area contains a table with the following columns: Company, Code, Description, Last Number, Start of Range, and End of Range. The table lists various system variables such as CASH_RC#, EX\$FILE, CRYST, APTBBATCH, APTBEXP#, ARIFILESEQ, ARSUSPITH, ASSET-NUM, ASSETNUM, AULFNDWK, AULPHSWK, BATCHID#, BDISBHNO, BDSYNUM, BEHDRTEXT#, and BITMTEXT#.

Company	Code	Description	Last Number	Start of Range	End of Range
	CASH_RC#	ML2	50000	1	99999999
	EX\$FILE	MB2	50011	1	9999
000	CRYST	BPC	48	1	9999
000	APTBBATCH	TC1	50000	1	99999999
000	APTBEXP#	TC1	50000	1	99999999
000	ARIFILESEQ	TC1	50000	1	99999999
000	ARSUSPITH	TEST	50020	1	99999999
000	ASSET-NUM	FA0	50008	1	99999999
000	ASSETNUM	ASSETNUM	50000	1	99999999
000	AULFNDWK	MPG	51600	1	99999999
000	AULPHSWK	DG2	50107	1	99999999
000	BATCHID#	CTR	50015	1	99999999
000	BDISBHNO	BPC	107	1	99999999
000	BDSYNUM	GM1	20	1	99999999
000	BEHDRTEXT#	TC1	50000	1	99999999
000	BITMTEXT#	TC1	50000	1	99999999

Global Variable Inquiry/List Window (IL1)

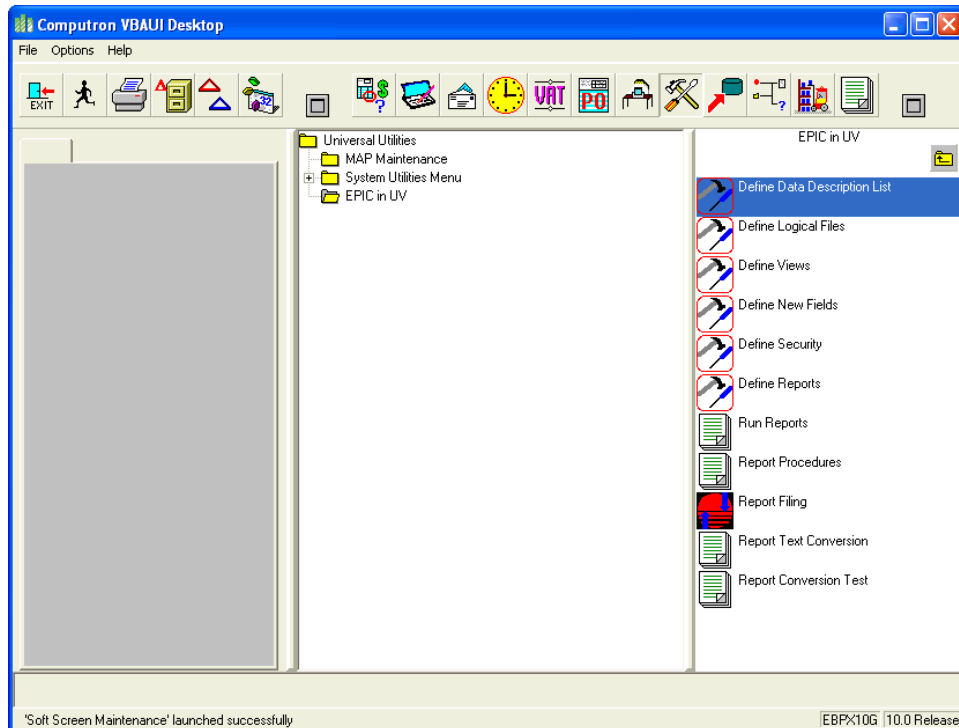
To display the details of a listed code, you can either:

- ◆ highlight it and click the Choose button;
- ◆ double-click the desired code.

1.9 EPIC in UV

Introduction to EPIC in UV

This function is used to access the menu for the EPIC Report & Query Generator. You can not only create customized versions of standard Computron EPIC reports, but also create your own reports/inquiries to access data in the Universal Utilities subsystem. For detailed information about this product, refer to the *EPIC Report & Query Generator* manual.

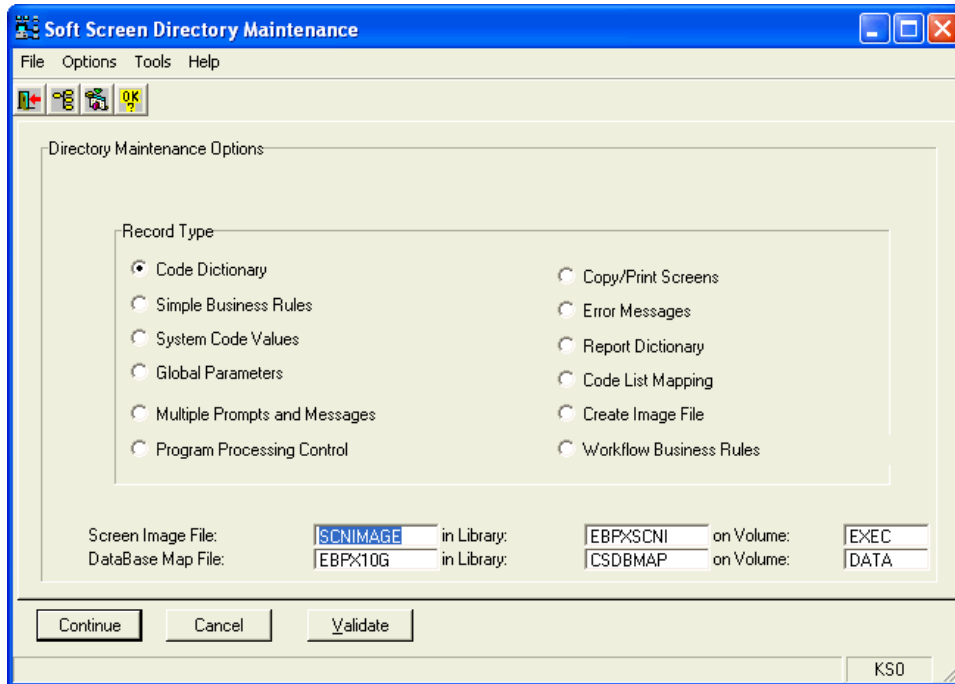


EPIC Report & Query Generator Menu

1.10 Soft Screen Maintenance

Introduction to Soft Screens Maintenance

This function is used to customize report formats to meet specific client requirements, and to effect global terminology changes. This module is available as a separate product offering from Computron. For detailed information about this product, refer to the *Soft Screens User's Guide*.



Soft Screen Maintenance Window (KSO)

1.11 Capture Log File to Print

Introduction to Capture Log File to Print

Printing out the log file is especially useful when analyzing system and/or program problems that produce error messages. Double-clicking this icon automatically generates a report containing the last 5,000 lines of your Computron Log file and places it in your print queue. A window does not display nor are you prompted for selection criteria.

Two files are written to your print queue. The user##### (where ##### represents an ordinal number based on previous versions of this file in your print queue) contains the selection criteria for the report. These selection criteria are not modifiable. The log file is written to a report titled logf#####, which is located in your print queue on the Spool directory as defined by the LGMAP.

Defining the Log File Size

You can set the default number of lines (N) that the Computron Log File Report extracts from the Computron Log file by accessing the LOCAL.START script and introducing the following statement:

```
export CTIRON_REPORT_LOG=${CTIRON_REPORT_LOG:-N}
```

Log File Report Description

Sample Computron Log File Report

```

LOG FILE = /enterprise/10qa/var/logs/user_pse
Retrieving last 5000 lines / Total lines = 25257
*****
PRNAME = MENUFILE   FILE = GLDIMREF, LIBRARY = CIGLMENU, VOLUME = IPLVOL
Read-s: 6           Read-g: 0           Read-d: 0           Read-nd: 0
Starts: 0           Writes: 0           Rewrites: 0         Deletes: 0

===== 04/10/07 14:08:58 PID: 06964 =====
RUNTIME: /enterprise/10qa/exec/_ctron/_wrunora(10.0.6.0) OS: SunOS(5.10)
PGM:  MENUFRT2(      )/CSACCESS      QUAL: Activity      TYPE: Totals
PRNAME = MENUFILE   FILE = GLPOSTAC, LIBRARY = CIGLMENU, VOLUME = IPLVOL
Read-s: 6           Read-g: 0           Read-d: 0           Read-nd: 0
Starts: 0           Writes: 0           Rewrites: 0         Deletes: 0

===== 04/10/07 14:08:58 PID: 06964 =====
RUNTIME: /enterprise/10qa/exec/_ctron/_wrunora(10.0.6.0) OS: SunOS(5.10)
PGM:  MENUFRT2(      )/CSACCESS      QUAL: Activity      TYPE: Totals
PRNAME = MENUFILE   FILE = GLTRPARM, LIBRARY = CIGLMENU, VOLUME = IPLVOL
Read-s: 6           Read-g: 0           Read-d: 0           Read-nd: 0
Starts: 0           Writes: 0           Rewrites: 0         Deletes: 0

===== 04/10/07 14:08:58 PID: 06964 =====
RUNTIME: /enterprise/10qa/exec/_ctron/_wrunora(10.0.6.0) OS: SunOS(5.10)
PGM:  MENUFRT2(      )/CSACCESS      QUAL: Activity      TYPE: Totals
PRNAME = MENUFILE   FILE = GLTRANS , LIBRARY = CIGLMENU, VOLUME = IPLVOL
Read-s: 6           Read-g: 0           Read-d: 0           Read-nd: 0
Starts: 0           Writes: 0           Rewrites: 0         Deletes: 0

===== 04/10/07 14:08:58 PID: 06964 =====
RUNTIME: /enterprise/10qa/exec/_ctron/_wrunora(10.0.6.0) OS: SunOS(5.10)
PGM:  MENUFRT2(      )/CSACCESS      QUAL: Activity      TYPE: Totals
PRNAME = MENUFILE   FILE = GLBUDGET, LIBRARY = CIGLMENU, VOLUME = IPLVOL
Read-s: 6           Read-g: 0           Read-d: 0           Read-nd: 0
Starts: 0           Writes: 0           Rewrites: 0         Deletes: 0

===== 04/10/07 14:08:58 PID: 06964 =====
RUNTIME: /enterprise/10qa/exec/_ctron/_wrunora(10.0.6.0) OS: SunOS(5.10)
PGM:  MENUFRT2(      )/CSACCESS      QUAL: Activity      TYPE: Totals
PRNAME = MENUFILE   FILE = GLTRGEN , LIBRARY = CIGLMENU, VOLUME = IPLVOL
Read-s: 6           Read-g: 0           Read-d: 0           Read-nd: 0
Starts: 0           Writes: 0           Rewrites: 0         Deletes: 0

===== 04/10/07 14:08:58 PID: 06964 =====
RUNTIME: /enterprise/10qa/exec/_ctron/_wrunora(10.0.6.0) OS: SunOS(5.10)
PGM:  MENUFRT2(      )/CSACCESS      QUAL: Activity      TYPE: Totals
PRNAME = MENUFILE   FILE = GLEDE , LIBRARY = CIGLMENU, VOLUME = IPLVOL
Read-s: 6           Read-g: 0           Read-d: 0           Read-nd: 0
Starts: 0           Writes: 0           Rewrites: 0         Deletes: 0

===== 04/10/07 14:08:58 PID: 06964 =====
RUNTIME: /enterprise/10qa/exec/_ctron/_wrunora(10.0.6.0) OS: SunOS(5.10)
PGM:  MENUFRT2(      )/CSACCESS      QUAL: Activity      TYPE: Totals
PRNAME = MENUFILE   FILE = GLPRW , LIBRARY = CIGLMENU, VOLUME = IPLVOL
Read-s: 6           Read-g: 0           Read-d: 0           Read-nd: 0
Starts: 0           Writes: 0           Rewrites: 0         Deletes: 0
    
```



```
===== 04/10/07 14:08:58 PID: 06964 =====
RUNTIME: /enterprise/10qa/exec/_ctron_/wrunora(10.0.6.0) OS: SunOS(5.10)
PGM: MENUPRT2( )/CSACCESS QUAL: Activity TYPE: Totals
PRNAME = MENUFILE FILE = GLLDGRPT, LIBRARY = CIGLMENU, VOLUME = IPLVOL
Read-s: 6 Read-g: 0 Read-d: 0 Read-nd: 0
Starts: 0 Writes: 0 Rewrites: 0 Deletes: 0

===== 04/10/07 14:08:58 PID: 06964 =====
RUNTIME: /enterprise/10qa/exec/_ctron_/wrunora(10.0.6.0) OS: SunOS(5.10)
PGM: MENUPRT2( )/CSACCESS QUAL: Activity TYPE: Totals
PRNAME = MENUFILE FILE = GLEXTRNL, LIBRARY = CIGLMENU, VOLUME = IPLVOL
Read-s: 6 Read-g: 0 Read-d: 0 Read-nd: 0
Starts: 0 Writes: 0 Rewrites: 0 Deletes: 0

===== 04/10/07 14:08:58 PID: 06964 =====
RUNTIME: /enterprise/10qa/exec/_ctron_/wrunora(10.0.6.0) OS: SunOS(5.10)
PGM: MENUPRT2( )/CSACCESS QUAL: Activity TYPE: Totals
PRNAME = MENUFILE FILE = GLREL, LIBRARY = CIGLMENU, VOLUME = IPLVOL
Read-s: 6 Read-g: 0 Read-d: 0 Read-nd: 0
Starts: 0 Writes: 0 Rewrites: 0 Deletes: 0

===== 04/10/07 14:08:58 PID: 06964 =====
RUNTIME: /enterprise/10qa/exec/_ctron_/wrunora(10.0.6.0) OS: SunOS(5.10)
PGM: MENUPRT2( )/CSACCESS QUAL: Activity TYPE: Totals
PRNAME = MENUFILE FILE = GLSYSADM, LIBRARY = CIGLMENU, VOLUME = IPLVOL
Read-s: 6 Read-g: 0 Read-d: 0 Read-nd: 0
Starts: 0 Writes: 0 Rewrites: 0 Deletes: 0

===== 04/10/07 14:08:58 PID: 06964 =====
RUNTIME: /enterprise/10qa/exec/_ctron_/wrunora(10.0.6.0) OS: SunOS(5.10)
PGM: MENUPRT2( )/CSACCESS QUAL: Activity TYPE: Totals
PRNAME = MENUFILE FILE = ENCMENU, LIBRARY = CIGLMENU, VOLUME = IPLVOL
Read-s: 6 Read-g: 0 Read-d: 0 Read-nd: 0
Starts: 0 Writes: 0 Rewrites: 0 Deletes: 0
```

1.12 Lockout Utility

Introduction to Lockout Utility

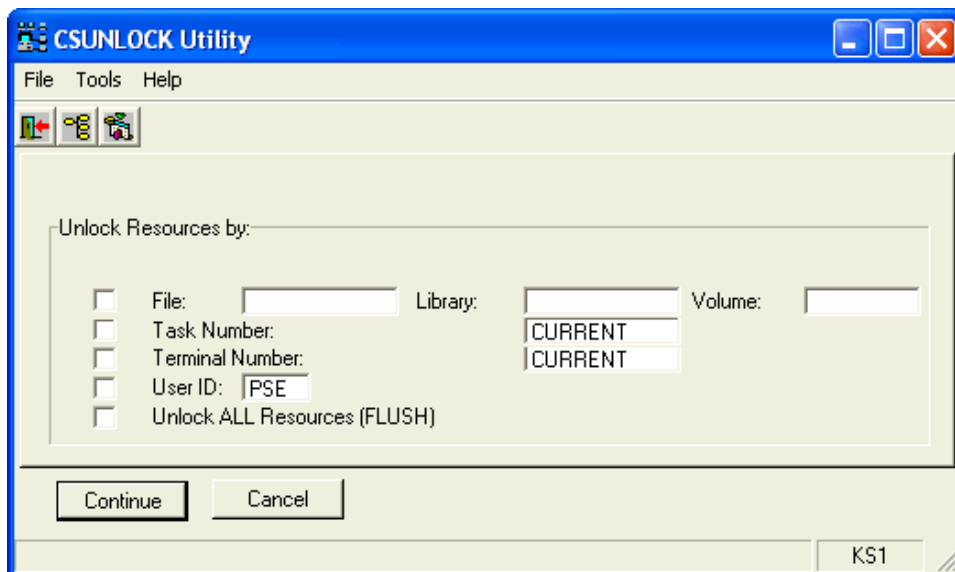
When a program needs to update tables, resources are held. This is necessary so that two programs competing for the same resource don't enter into an infinite lock and halt processing. In normal processing, resources are released when the function is complete. In case the system malfunctions, held resources may not get released and other functioning cannot proceed. This function is used to release (unlock) the affected resource(s) by:

- ◆ File/Library/Volume
- ◆ Task Number
- ◆ Terminal Number
- ◆ User ID
- ◆ Unlock ALL Resources (FLUSH).

⚠ *Warning: Unlock ALL Resources (FLUSH) releases all users' resources. This option should be used with great caution and only if you know you are not disrupting other users' processing.*

CSUNLOCK Utility Window (KS1)

The following window displays when you access the Lockout Utility function:



CSUNLOCK Utility Window (KS1)

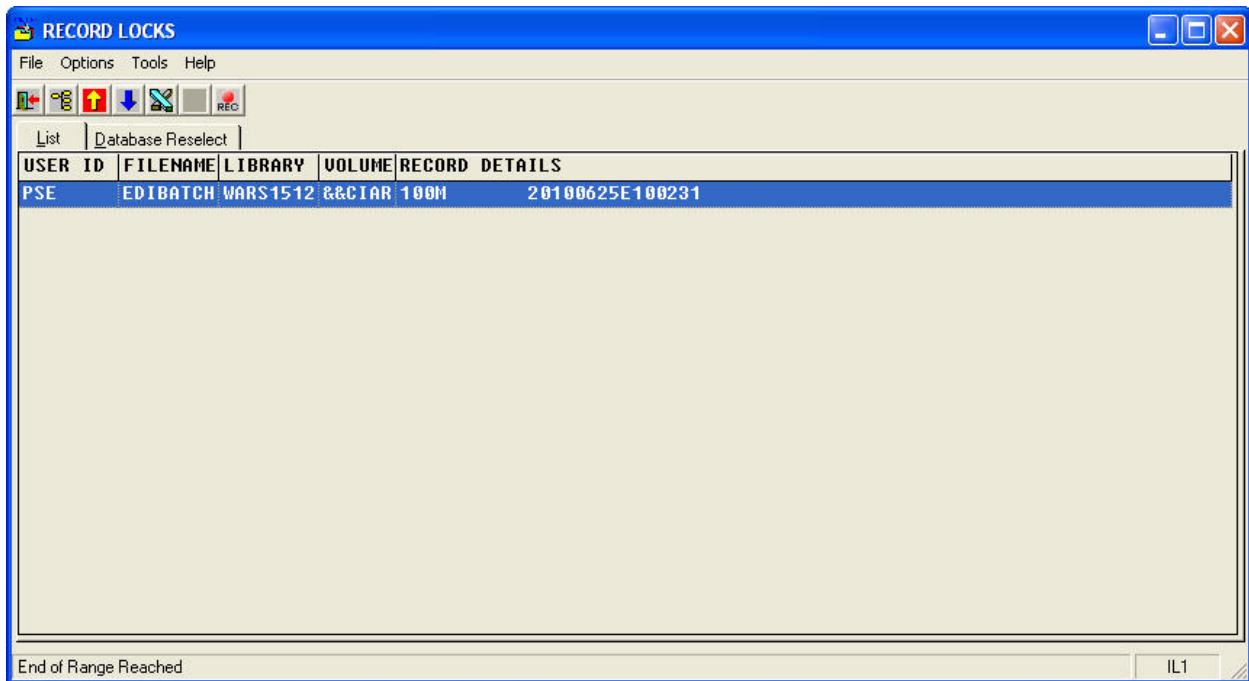
1.13 Lockout Inquiry

Introduction to Lockout Inquiry

This inquiry allows you to see what records are currently locked in the system. In order to unlock these records, you must use the Lockout Utility or Lockout Delete by User functions which are also found on the System Utilities Menu.

Record Locks Window (IL1)

The following window displays when you access the Lockout Inquiry function: It lists all records that are currently on hold.



Record Locks Window (KS1)

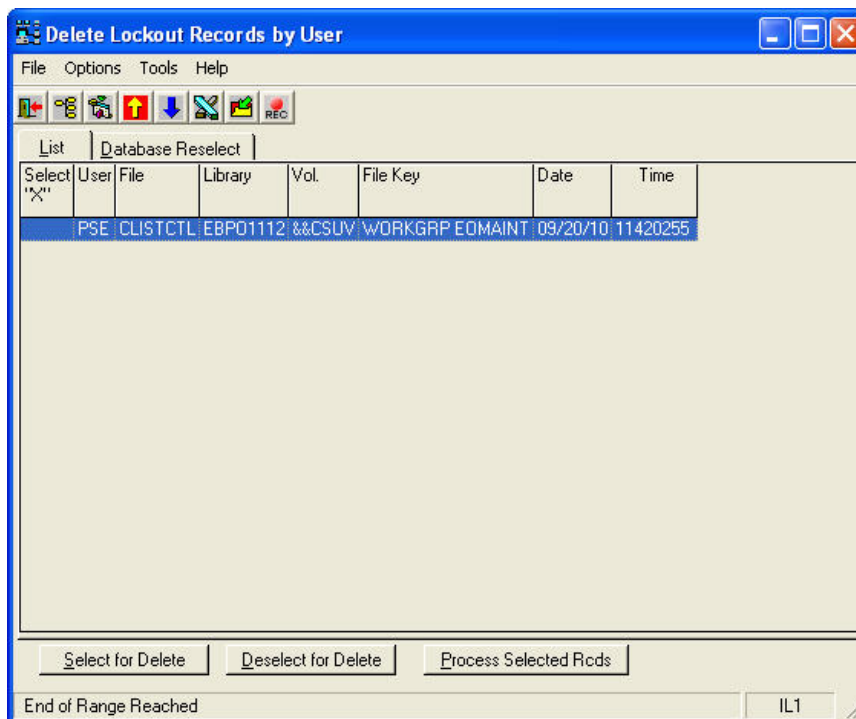
1.14 Lockout Delete by User

Introduction to Lockout Delete by User

This tool allows you to see what records are locked within the system and then unlock selected holds. In contrast, the Lockout Utility function does not provide the ability to remove individual lockouts.

Delete Lockout Records by User Window (IL1)

The following window displays when you access the Lockout Delete by User function: It lists all records that are currently locked.



Delete Lockout Records by User Window (KS1)

Option	Description
Select for Delete	Highlight one or more records to be selected for unlocking and click this option. An "X" is placed in the Select "X" column. If there are any that should not have been selected, use the Deselect for Delete button to remove the X from the Select "X" column.
Deselect for Delete	Highlight one or more records to be deselected from the group to be unlocked and click this option. The "X" in the Select "X" column is removed.
Process Selected Records	This unlocks the records that have been selected and returns you to the menu.

Option	Description
Exit or Previous	This exits the function and returns you to the menu. Any records selected for unlocking are not processed.
More Records	This accesses the next group of records from the database.
X (Excel)	Download the data to an Excel spreadsheet.

1.15 System Access Inquiry/List

Introduction to System Access Inquiry/List

This function provides a means to report on system access in a format that can be imported into any application, such as a spreadsheet. The program creates a file that contains both the users and the Computron menus and/or programs to which they have access. The file contains security information from the menu and user files. Selection criteria allow you to limit the information by a range of users or user groups. Additionally, data can be sorted by a hierarchy that you establish at run time.

Most of the information in the file is displayed or printed, one line per user or group. Any information on the file that is not already displayed or printed is available by modifications to the “soft” screens and reports.

This function can be used to provide proof that the company is in compliance with security measures.

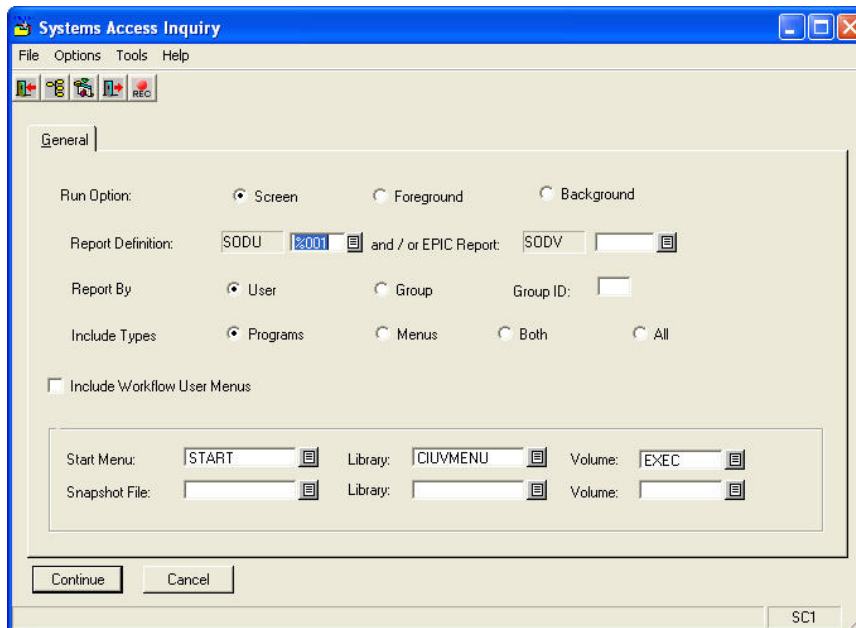
How the Information for the Report is Determined

All users who have access to Computron applications are identified by the system via a user account. The user account establishes system-wide security privileges for the user. It also assigns the user to a user group.

At the menu level, the user’s access to menu options is determined by the security level of the user compared to the security level of the menu option. Where the security level of the menu option is higher than the user’s security level, the menu option does not display on the user’s menu. There is also specific security where the menu option can allow or disallow a user based on the user ID or the group ID.

System Access Inquiry Window (SC1)

On this window, set parameters for the type of data to be accessed and how it is presented.



System Access Inquiry Window (SC1)

Run Option

This is the option used to run this report. Valid options are:

Screen – This produces the report in foreground and presents it online.

Foreground – Process in foreground. Use of this option submits the job to processing immediately.

Background – Process in background. Use of this option submits the job to a background queue, where it is processed as resources become available.

Default: Screen.

Report Definition

This is the soft report to produce. This only applies when the Run Option is set to Foreground or Background. Both an EPIC report and/or this standard report can be printed in the same run.

☞ *Note:* If you want to tailor a report, the report must start with SODU for users and SODG for groups.

Default: SODU%001.

and / or EPIC Report

This is the EPIC report to produce. This only applies when the Run Option is set to Foreground or Background. Both a standard report and/or this EPIC report can be printed in the same run.

☞ *Note:* If you want to tailor a report, the report must start with SODU for users and SODG for groups.

Report By

This determines the value that the report is based on. Valid options are:

User – Base the report on user IDs.

Group – Base the report on the group ID indicated in the Group ID field.

Default: User.

Group ID

This identifies the group for the report. This is used when the Report By field is set to Group.

Include Types

This determines what type of information is included from the Menu files. Valid options are:

Programs – Include only those programs that meet the criteria established above.

Menus – Include only those menus that meet the criteria established above.

Both – Include both programs and menus. This excludes comments, exits and the Run Any Program option.

All – Include all information from the from the Menu files. This includes programs, menus, comments, exits, and the Run Any Program option.

Default: Programs.

Include Workflow User Menus

This determines whether the Workflow menus for individual users are included in the output. This only applies if Include Types is not set to Programs.

Valid options are:

Include menus created for individual users via the Workflow system.

Do not include Workflow user menus.

Default:

Start Menu Library Volume

This identifies the menu from which to start reporting.

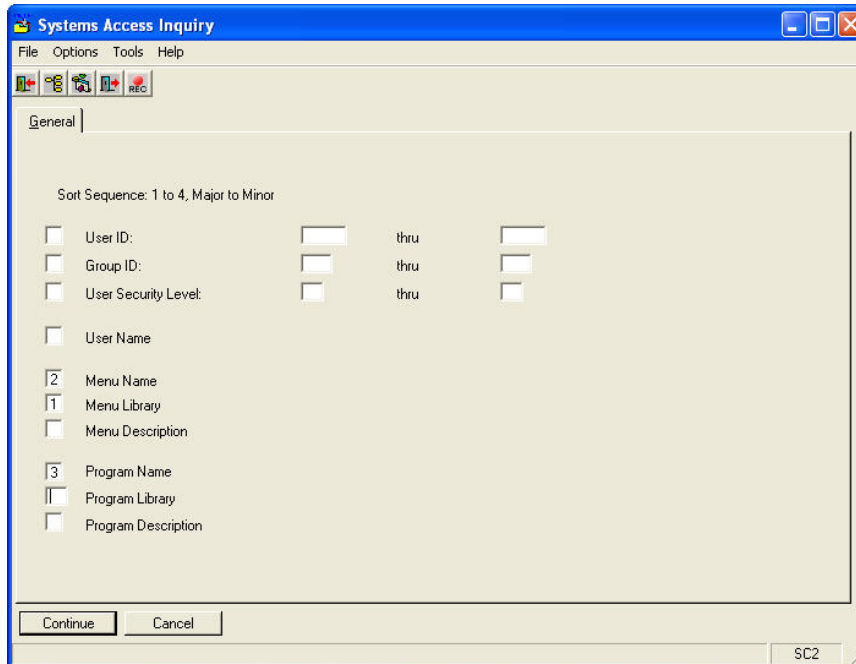
☞ *Note:* Due to the extremely large number of menu options in the full Computron system, use this field to limit the extent of the report.

Snapshot File Library Volume

Specify the name and location of the file, based on the input criteria, that will be created. If this file already exists, you can overwrite the file by clicking OK in the pop-up window. The file created is an indexed file, SODUWORK.

System Access Inquiry Window (SC2)

This window allows you to establish the range of user IDs and/or group IDs and the order to sort the information.



System Access Inquiry Window (SC2)

Sort Sequence

Use the boxes on the left side of the window to identify the sort sequence for your report. Enter the ascending sort sequence to determine the order of the extracted information. The order should be entered from 1 to 4; with 1 being the major sort and 4 is the maximum number of sorts. In the example above, one user is sorted by Menu Library, Menu Name and Program Name.

If the sort order is not entered, the sort defaults. When reporting by user, the sort order defaults to User ID followed by Group ID. If reporting by group, the user fields on the SC2 window are not available and the sort order defaults to Group ID.

User ID thru

Enter a range of user IDs to be included in the report. This range is not available if reporting by Group.

Note: It is not recommended to leave this range blank. The system needs to cycle through all menus from the indicated Start Menu for every user in the User file. The report produced may take a long time and be extremely large.

Group ID thru

Enter a range of group IDs to be included in the report. When reporting by group, this range defaults to the entry on the SC1 window.

User Security Level thru

Enter a range of security levels to be included in the report.

System Access Inquiry Window (IL1 and IL1 G)

These windows display the information requested when the Run Option is set to Screen. The IL1 window displays when the Report By field is set to Users. The IL1 G window displays when the Report By field is set to Group.

User ID	User Name	Group ID	User SL	Program Name	Program Library	Program Description	Prog Type	Prog SL	Menu Name	Menu Library	Menu Description	Sort Key
PSE	Pat Eckart-McGovern	9	CSDISPLY	CSUVDLOAD	CSUVDLOAD	Display Utility	P	9	APADJMU	CIAPMENU	Adjustment Authorization Codes	CIAPMENU APADJMU CSDISPLY
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVDLOAD	CSUVDLOAD	Adjustment Authorization Maint	P	0	APADJMU	CIAPMENU	Adjustment Authorization Codes	CIAPMENU APADJMU CSENTRY
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVDLOAD	CSUVDLOAD	Adj Authorization Inq/List	P	0	APADJMU	CIAPMENU	Adjustment Authorization Codes	CIAPMENU APADJMU CSENTRY
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVDLOAD	CSUVDLOAD	Adjustment Reason Maint	P	0	APADJMU	CIAPMENU	Adjustment Authorization Codes	CIAPMENU APADJMU CSENTRY
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVDLOAD	CSUVDLOAD	Adjustment Reason Inq/List	P	0	APADJMU	CIAPMENU	Adjustment Authorization Codes	CIAPMENU APADJMU CSENTRY
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVDLOAD	CSUVDLOAD	Discount Lost Reason Maint	P	0	APADJMU	CIAPMENU	Adjustment Authorization Codes	CIAPMENU APADJMU CSENTRY
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVDLOAD	CSUVDLOAD	Disc Lost Reason Inq/List	P	0	APADJMU	CIAPMENU	Adjustment Authorization Codes	CIAPMENU APADJMU CSENTRY
PSE	Pat Eckart-McGovern	9	@QUEUE?			your Print Queue	P	0	APAUDIT	CIAPMENU	A/P Audit	CIAPMENU APAUDIT @QUEUE?
PSE	Pat Eckart-McGovern	9	AUDAPPRO	CIAPRUN	CIAPRUN	A/P Audit Reports	P	0	APAUDIT	CIAPMENU	A/P Audit	CIAPMENU APAUDIT AUDAPPRO
PSE	Pat Eckart-McGovern	9	CSDISPLY	CSUVDLOAD	CSUVDLOAD	Display Utility	P	9	APAUDIT	CIAPMENU	A/P Audit	CIAPMENU APAUDIT CSDISPLY
PSE	Pat Eckart-McGovern	9	HST10990	CIAPLOAD	CIAPLOAD	1099 Verification Report	P	0	APAUDIT	CIAPMENU	A/P Audit	CIAPMENU APAUDIT HST10990
PSE	Pat Eckart-McGovern	9	VREMHIL0	CIAPLOAD	CIAPLOAD	Payee History Inq/List	P	0	APAUDIT	CIAPMENU	A/P Audit	CIAPMENU APAUDIT VREMHIL0
PSE	Pat Eckart-McGovern	9	VREMRPT0	CIAPLOAD	CIAPLOAD	Payee History Comparison List	P	0	APAUDIT	CIAPMENU	A/P Audit	CIAPMENU APAUDIT VREMRPT0
PSE	Pat Eckart-McGovern	9	@QUEUE?			your Print Queue	P	0	APDISTRG	CIAPMENU	Register Reports	CIAPMENU APDISTRG @QUEUE?
PSE	Pat Eckart-McGovern	9	APREGPP0	CIAPLOAD	CIAPLOAD	Register Report	P	0	APDISTRG	CIAPMENU	Register Reports	CIAPMENU APDISTRG APREGPP0
PSE	Pat Eckart-McGovern	9	APREGPT0	CIAPLOAD	CIAPLOAD	Register Query Report	P	0	APDISTRG	CIAPMENU	Register Reports	CIAPMENU APDISTRG APREGPT0
PSE	Pat Eckart-McGovern	9	APZOOM	CIAPLOAD	CIAPLOAD	ZOOM Processing	P	0	APDISTRG	CIAPMENU	Register Reports	CIAPMENU APDISTRG APZOOM
PSE	Pat Eckart-McGovern	9	CDSTPT0	CIAPLOAD	CIAPLOAD	Statement of Disbursements	P	0	APDISTRG	CIAPMENU	Register Reports	CIAPMENU APDISTRG CDSTPT0

Segregation of Duties Inquiry Window (IL1)

Group ID	Program Name	Program Library	Program Description	Prog Type	Prog SL	Menu Name	Menu Library	Menu Description	Sort Key
TB	CSDISPLY	CSUVDLOAD	Display Utility	P	9	APADJMU	CIAPMENU	Adjustment Authorization Codes	CIAPMENU APADJMU CSDISPLY
TB	CSENTRY	CSUVDLOAD	Adjustment Authorization Maint	P	0	APADJMU	CIAPMENU	Adjustment Authorization Codes	CIAPMENU APADJMU CSENTRY
TB	CSENTRY	CSUVDLOAD	Adj Authorization Inq/List	P	0	APADJMU	CIAPMENU	Adjustment Authorization Codes	CIAPMENU APADJMU CSENTRY
TB	CSENTRY	CSUVDLOAD	Adjustment Reason Maint	P	0	APADJMU	CIAPMENU	Adjustment Authorization Codes	CIAPMENU APADJMU CSENTRY
TB	CSENTRY	CSUVDLOAD	Adjustment Reason Inq/List	P	0	APADJMU	CIAPMENU	Adjustment Authorization Codes	CIAPMENU APADJMU CSENTRY
TB	CSENTRY	CSUVDLOAD	Discount Lost Reason Maint	P	0	APADJMU	CIAPMENU	Adjustment Authorization Codes	CIAPMENU APADJMU CSENTRY
TB	CSENTRY	CSUVDLOAD	Disc Lost Reason Inq/List	P	0	APADJMU	CIAPMENU	Adjustment Authorization Codes	CIAPMENU APADJMU CSENTRY
TB	@QUEUE?		your Print Queue	P	0	APAUDIT	CIAPMENU	A/P Audit	CIAPMENU APAUDIT @QUEUE?
TB	AUDAPPRO	CIAPRUN	A/P Audit Reports	P	0	APAUDIT	CIAPMENU	A/P Audit	CIAPMENU APAUDIT AUDAPPRO
TB	CSDISPLY	CSUVDLOAD	Display Utility	P	9	APAUDIT	CIAPMENU	A/P Audit	CIAPMENU APAUDIT CSDISPLY
TB	HST10990	CIAPLOAD	1099 Verification Report	P	0	APAUDIT	CIAPMENU	A/P Audit	CIAPMENU APAUDIT HST10990
TB	VREMHIL0	CIAPLOAD	Payee History Inq/List	P	0	APAUDIT	CIAPMENU	A/P Audit	CIAPMENU APAUDIT VREMHIL0
TB	VREMRPT0	CIAPLOAD	Payee History Comparison List	P	0	APAUDIT	CIAPMENU	A/P Audit	CIAPMENU APAUDIT VREMRPT0
TB	@QUEUE?		your Print Queue	P	0	APDISTRG	CIAPMENU	Register Reports	CIAPMENU APDISTRG @QUEUE?
TB	APREGPP0	CIAPLOAD	Register Report	P	0	APDISTRG	CIAPMENU	Register Reports	CIAPMENU APDISTRG APREGPP0
TB	APREGPT0	CIAPLOAD	Register Query Report	P	0	APDISTRG	CIAPMENU	Register Reports	CIAPMENU APDISTRG APREGPT0
TB	APZOOM	CIAPLOAD	ZOOM Processing	P	0	APDISTRG	CIAPMENU	Register Reports	CIAPMENU APDISTRG APZOOM
TB	CDSTPT0	CIAPLOAD	Statement of Disbursements	P	0	APDISTRG	CIAPMENU	Register Reports	CIAPMENU APDISTRG CDSTPT0

Segregation of Duties Inquiry Window (IL1 G)

System Access Report Definition

System Access Report

```

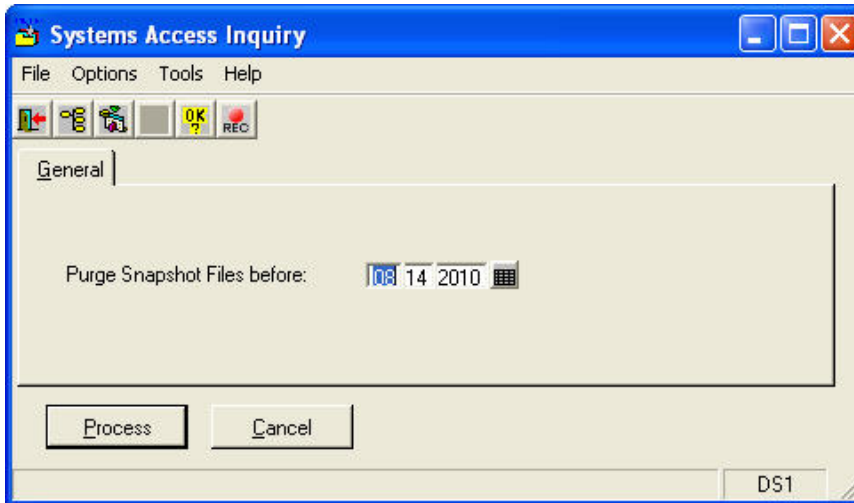
*
* ***** Mon Sep 13, 2010 2:45 PM *
*
* Systems Access Report
*
* ***** Page 3 *

```

USER		PROGRAM			MENU				
ID	Name	Grp	SL	Name / Library	Description	Typ	SL	Name / Library	Description
PSE	Pat Eckart-McGovern	9	CSDISPLY	CSUVLOAD	Display Utility	P	9	APADJMU	CIAPMENU Adjustment Authorization Co
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVLOAD	Adjustment Authorization	P	0	APADJMU	CIAPMENU Adjustment Authorization Co
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVLOAD	Adj Authorization Inq/Lis	P	0	APADJMU	CIAPMENU Adjustment Authorization Co
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVLOAD	Adjustment Reason Maint	P	0	APADJMU	CIAPMENU Adjustment Authorization Co
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVLOAD	Adjustment Reason Inq/Lis	P	0	APADJMU	CIAPMENU Adjustment Authorization Co
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVLOAD	Discount Lost Reason Main	P	0	APADJMU	CIAPMENU Adjustment Authorization Co
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVLOAD	Disc Lost Reason Inq/List	P	0	APADJMU	CIAPMENU Adjustment Authorization Co
PSE	Pat Eckart-McGovern	9	@QUEUE?		your Print Queue	P	0	APAUDIT	CIAPMENU A/P Audit
PSE	Pat Eckart-McGovern	9	AUDAPPRO	CIAPRUN	A/P Audit Reports	P	0	APAUDIT	CIAPMENU A/P Audit
PSE	Pat Eckart-McGovern	9	CSDISPLY	CSUVLOAD	Display Utility	P	9	APAUDIT	CIAPMENU A/P Audit
PSE	Pat Eckart-McGovern	9	HST10990	CIAPLOAD	1099 Verification Report	P	0	APAUDIT	CIAPMENU A/P Audit
PSE	Pat Eckart-McGovern	9	VREMH110	CIAPLOAD	Payee History Inq/List	P	0	APAUDIT	CIAPMENU A/P Audit
PSE	Pat Eckart-McGovern	9	VREMRPT0	CIAPLOAD	Payee History Comparison	P	0	APAUDIT	CIAPMENU A/P Audit
PSE	Pat Eckart-McGovern	9	@QUEUE?		your Print Queue	P	0	APDISTRG	CIAPMENU Register Reports
PSE	Pat Eckart-McGovern	9	APREGPP0	CIAPLOAD	Register Report	P	0	APDISTRG	CIAPMENU Register Reports
PSE	Pat Eckart-McGovern	9	APREGPT0	CIAPLOAD	Register Query Report	P	0	APDISTRG	CIAPMENU Register Reports
PSE	Pat Eckart-McGovern	9	APZOOM	CIAPLOAD	ZOOM Processing	P	0	APDISTRG	CIAPMENU Register Reports
PSE	Pat Eckart-McGovern	9	CDSTPT0	CIAPLOAD	Statement of Disbursement	P	0	APDISTRG	CIAPMENU Register Reports
PSE	Pat Eckart-McGovern	9	CSDISPLY	CSUVLOAD	Display Utility	P	9	APDISTRG	CIAPMENU Register Reports
PSE	Pat Eckart-McGovern	9	APBKFMNT	CIAPLOAD	Bank Data Format Maint	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	CHKGEN0	CIAPLOAD	GENEX-In Non-AP Checks	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	CKZTAPP0	CIAPLOAD	GENEX-Out checks	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	CSDISPLY	CSUVLOAD	Display	P	9	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	CTLMNT2	CSUVLOAD	Data Description List Mai	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	EXT1099	CIAPRUN	GENEX-Out 1099 Print	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	FGEN0	CIAPLOAD	GENEX-In Factors	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	FORM1042	CIAPLOAD	GENEX-Out 1042 Print	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	ICGEN0	CIAPLOAD	GENEX-In Vouchers	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	TAP1042	CIAPLOAD	GENEX-Out Vendor 1042 Dat	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	TAP1099	CIAPLOAD	GENEX-Out Vendor 1099 Dat	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	TAPETOCK	CIAPLOAD	GENEX-In Check Reconcilia	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	VENDGEN0	CIAPLOAD	GENEX-In Vendors	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	VOUCHUP0	CIAPLOAD	GENEX-Out Vouchers	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	XVOUCHG0	CIAPLOAD	GENEX-In Voucher Changes	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	CSDISPLY	CSUVLOAD	Display Utility	P	9	APFINMU	CIAPMENU Financial Codes Approval
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVLOAD	AP Financial Approval ID	P	0	APFINMU	CIAPMENU Financial Codes Approval
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVLOAD	AP Fin Approval ID Inq/Li	P	0	APFINMU	CIAPMENU Financial Codes Approval
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVLOAD	AP Payment Approval Maint	P	0	APFINMU	CIAPMENU Financial Codes Approval
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVLOAD	AP Payment Approval Inq/L	P	0	APFINMU	CIAPMENU Financial Codes Approval
PSE	Pat Eckart-McGovern	9	@QUEUE?		your Print Queue	P	0	APGST	CIAPMENU GST
PSE	Pat Eckart-McGovern	9	APREGPP0	CIAPLOAD	GST Posting Update	P	0	APGST	CIAPMENU GST
PSE	Pat Eckart-McGovern	9	APREGPPR	CIAPRUN	GST Posting Edit	P	0	APGST	CIAPMENU GST
PSE	Pat Eckart-McGovern	9	CSDISPLY	CSUVLOAD	Display Utility	P	9	APGST	CIAPMENU GST
PSE	Pat Eckart-McGovern	9	CSENTRY	CSUVLOAD	Non-Rebateable Acct Maint	P	0	APGST	CIAPMENU GST
PSE	Pat Eckart-McGovern	9	CSEPIC0	CIRPLOAD	Non-Rebateable Acct List	P	0	APGST	CIAPMENU GST
PSE	Pat Eckart-McGovern	9	CSEPIC0	CIRPLOAD	Exception Account List	P	0	APGST	CIAPMENU GST

Purge Snapshot Files

This option is available on the File menu. It allows you to remove snapshot files from the server before a specified date. The default date is one month prior to today's date.



System Access Inquiry Window (DS1)

Purge Snapshot Files before

Enter the date to determine which files to purge.

Default: One month before the current system date.

1.16 Print Audit File

Introduction to Print Audit File

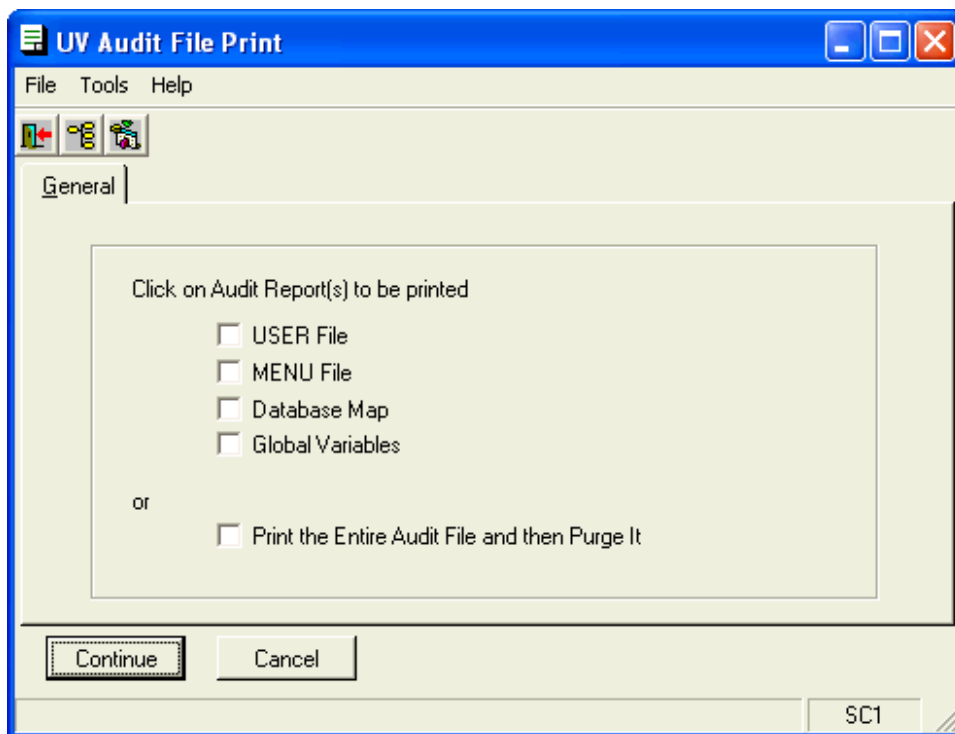
This function gives you the option to print a report on the following files:

- ◆ User file (maintained via User File Maintenance);
- ◆ Menu file (maintained via Menu File Maintenance);
- ◆ Database Map file (maintained via Database Map Maintenance);
- ◆ Global Variable file (maintained via Global Variable Maintenance);
- ◆ All of the above (Audit file is subsequently purged).

Print Audit File Window (SC1)

The Print Audit File Window (SC1) displays when you select the Print Audit Report function.

Choose the information to include in the Audit report (User file, Menu file, Database Map file, Global Variable file) by selecting the appropriate option. You may also elect to include all of the above information in the Audit file and to then purge it.



Print Audit Report Window

Audit Report Description

The following sample reports are available from the Print Audit Report function. Each request produces an Audit Report Request (audu####) along with the requested audit file report(s).

Audit Report Request (audu####)

```

*                               ***** Mon Apr 16, 2007  8:52 AM *
*
*
*                               ***** Page 1 *
UV Audit File Print
-----
Click on Audit Report(s) to be printed
  USER File
  MENU File
  X Database Map
  Global Variables
or
  Print the Entire Audit File and then Purge It
CAS/IV Version: 10.00.00  DBMAP: EBPX10G  CSDBMAP  DATA

```

User File Option (user####)

```

*   Computron Software, LLC                               ***** Mon Apr 16, 2007  8:48 AM *
*
*                               UNIVERSAL FILE AUDIT REPORT
*                               USER FILE
*
*                               ***** Page 1 *
Audit Data --- Mode: B  User: VS  Date: 04/11/07  Time: 11:45
  User ID: VS          User Name: Vic Surajballi
  Open File: FINAL90  Open Lib: CSDBMAP      Open Vol: DATA    Op Lvl: 9    Sec Lvl: 9
  Prntr Cls:          Pr Mode: H          Prntr Nbr: 001     Comp (1 - 9):
  Def Comp: 110      Corp Sw: U          A/D Comp Sw:      Guest Passwd:
  Grp Code:          Guest ID:
-----
Audit Data --- Mode: C  User: VS  Date: 04/11/07  Time: 11:45
  User ID: VS          User Name: Vic Surajballi
  Open File: EBPX10G  Open Lib: CSDBMAP      Open Vol: DATA    Op Lvl: 9    Sec Lvl: 9
  Prntr Cls:          Pr Mode: H          Prntr Nbr: 001     Comp (1 - 9):
  Def Comp: 100      Corp Sw: U          A/D Comp Sw:      Guest Passwd:
  Grp Code:          Guest ID:
-----
Audit Data --- Mode: B  User: EDV Date: 04/11/07  Time: 11:51
  User ID: EDV        User Name: Earl Voss
  Open File: FINAL90  Open Lib: CSDBMAP      Open Vol: DATA    Op Lvl: 9    Sec Lvl: 9
  Prntr Cls: A        Pr Mode: K          Prntr Nbr: 003     Comp (1 - 9):
  Def Comp:          Corp Sw: U          A/D Comp Sw:      Guest Passwd:
  Grp Code:          Guest ID:
-----
Audit Data --- Mode: C  User: EDV Date: 04/11/07  Time: 11:51
  User ID: EDV        User Name: Earl Voss
  Open File: WFGL10G  Open Lib: CSDBMAP      Open Vol: DATA    Op Lvl: 9    Sec Lvl: 9
  Prntr Cls: A        Pr Mode: K          Prntr Nbr: 003     Comp (1 - 9):
  Def Comp:          Corp Sw: U          A/D Comp Sw:      Guest Passwd:
  Grp Code:          Guest ID:

```

Menu File Option (menu####)

```

*   Computron Software, LLC                               ***** Mon Apr 16, 2007  8:57 AM *
*
*                               UNIVERSAL FILE AUDIT REPORT
*                               MENU FILE
*
*                               ***** Page 1 *
Audit Data --- Mode:      User: PSE  Date: 04/16/07  Time: 08:56
Menu File --- Name: APTRANS  Library: CIAPMENU  Volume: EXEC

  Comp Name:                               Tab Allow: Y
  Sys Name:                               Disp Form: T
  Menu Name: Transaction Entry and Process  Disable BS: N
Line Nbr:  PF Name---- Library- Volume T H S Description-----
Line  1:   00                               X N 0 Voucher Entry --
Line  2:   01 APBATMT0  CIAPLOAD      P N 0 Batch Control Maintenance
Line  3:   02 ICENTRY  CIAPLOAD      P N 0 Voucher Entry
Line  4:   03 ICENTRY  CIAPLOAD      P N 0 Quick Voucher Entry
Line  5:   00                               0
Line  6:   00                               X N 0 Voucher Edit --
Line  7:   04 ICCHANG  CIAPLOAD      P N 0 Voucher Change
Line  8:   05 ICTRANS  CIAPLOAD      P N 0 Voucher Transfer/Rev/Del
Line  9:   06 VOUHRMNT CIAPLOAD      P N 0 Hold/Release Maintenance
Line 10:   00                               0
Line 11:   00                               0
Line 12:   07 APRECUR  CIAPMENU      M N 0 Recurring Transactions
Line 13:   00                               0
Line 14:   00                               0
Line 15:   00                               0
Line 16:   00                               0
Line 17:   00                               X N 0 Miscellaneous Entry --
Line 18:   08 REGENCY  CIAPLOAD      P N 0 Miscellaneous Entry/Edit
Line 19:   16                               E N 0 Return to Prior Menu
Line 20:   17                               U N 9 Run any Program
Line 21:   18 CSDISPLY CSUVLOAD      P N 9 Display Utility
Line 22:  EN/2 @QUEUE?                  p N 0 your Print Queue

```

Database Map Option (dbmp####)

```

*   Computron Software, LLC   ***** Mon Apr 16, 2007  8:52 AM *
*
*   UNIVERSAL FILE AUDIT REPORT
*   DBMAP FILE
*
***** Page 1 *

```

```

Audit Data --- Mode: C   User: PSE   Date: 04/10/07   Time: 16:18
DB Map --- Name: WPAR10G   Library: CSDBMAP   Volume: DATA

```

Log Libr	Phy Libr	Volume	Log File	Act File
&AUDT	CSUTAUDT	DATA		
&CTL		EXEC		
&CTLI		EXEC		
&EPIC		EXEC		
&MAP		10QMAP		
&PROC	&RUN	EXEC		
&PROG	&LOAD	EXEC		
&SCNI		EXEC	SCNIMAGE	SCNIMAGE
@RDBNAME	QA10G		QA10D	YY
CIAP	QA10D	&&CIAP		
CIAP	WHAPMS	DATA		
CIAP	&WORK		APREGPSI	
CIAP	QA10D	&&CIGL	AUTOBAL	
CIAP	&WORK	WRKVOL	EDETRANS	&UT#EDE
CIAP	&WORK	WRKVOL	PAYVOU	
CIAP	CIAPDATA	DATA	VOUCHER	VOUCHPSE
CIAR	QA10D	&&CIAR		
CIAR	WHARMS	DATA		
CIAR	QA10D	&&CIGL	AUTOBAL	
CIDI	QA10D	&&CIDI		
CIDI	WHIDMS	DATA		
CIPA	QA10D	&&CIPA		
CIPA	WHFAMS	DATA		
CIGL	QA10D	&&CIGL		
CIGL	WHGLMS	DATA		
CIGL	WHGLMS	DATA	EDETRANS	&UT#EDF
CIGL	CIGLWORK	DATA	GLTRNFMT	&UT#FMT
CIWF	QA10D	&&CIWF		
CIWF	WHVTMS	DATA		
CIWF	QA10D	&&CIWF		
CIWF	WFAR10G	DATA		
CIWF	BPAL0	&&CIWF	CONTACT	CLISTCTL
CIWF	WFAR10G	DATA	WFTASK	CLISTCTL
CIWFCTL	WFARCTL	EXEC	SYSVAR	SYSVAR
CIWFMENU	WFARMENU	EXEC		
CSUV	QA10D	&&CSUV		
CSUV	WHUVMS	DATA		
CSUV	QA10D	&&TEXT	TEXT	
CSUV	_CTRON_	EXEC	USERFILE	USERFIL3
NOTEFILE	\$DAY	010003		
SCANVOL	\$DAY	010001		
UTILMENU	CIWFMENU	EXEC		
WPAD	BPAL0	&&WPAD		
WPAD	BPAL0	&&CIWF	CONTACT	CLISTCTL
WFAR	QA10D	&&WFAR		
WFAR	WFAR10G	DATA		
WFAR	QA10D	&&WFAR	CLISTCST	CLISTCST
WFAR	QA10D	&&CIWF	CLISTCTL	CLISTCTL
WFAR	BPAL0	&&CIWF	CONTACT	CLISTCTL
WFAR	QA10D	&&CIWF	IMAGEFIL	IMAGEFIL

Global Variables Option (glbv####)

```

*   Computron Software, LLC   ***** Mon Apr 16, 2007  8:54 AM *
*
*   UNIVERSAL FILE AUDIT REPORT
*   GLOBAL VARIABLES
*
***** Page 1 *

```

```

Audit Data --- Mode: A   User: PSE   Date: 04/16/07   Time: 08:53
COMP: 003   CODE: VOU   DESC: VOUCHER NUMBER
START RNG: 1   END RNG: 999999998   LAST NUM: 50465

```

```

Audit Data --- Mode: D   User: PSE   Date: 04/16/07   Time: 08:53
COMP: 003   CODE: VOU   DESC: VOUCHER NUMBER
START RNG: 1   END RNG: 999999998   LAST NUM: 50465

```

Chapter 2

Security Subsystem

2.1 Security Subsystem

Computron's financial application software provides businesses in all major industries with a comprehensive solution for financial information management, addressing the needs of finance, management, audit and operations. As an integrated accounting system, Computron's financial application software is responsible for storing vital and confidential information for many businesses worldwide. In order to keep this information safe, Computron software provides an expansive security subsystem that complements the native OS (operating system) security, as well as the underlying RDBMS security.

Authentication

One of the major elements of security is authentication. In order to gain access to a Unix system, a relational database or the Computron applications, users must supply information that identifies them to the system. Access to the Computron applications begins with access to the Unix system where the Computron software resides. The user must log onto the Unix system and the Unix system must validate the user based on the Unix password file. The Unix password file contains one record for each user who is permitted access to the system. When a user attempts to logon, this file is checked to determine whether processing can continue. Additionally, the Unix user is identified to the system via a user account. The user account is linked to the password file for the purposes of authentication. The user account also establishes system-wide security privileges for the user such as assignment to a particular user group. The user ID and the user group determine the directories and files the user can access and the extent of that access (e.g., whether the user can modify or delete a file).

Once a user accesses the Unix system, the user must be validated within the Computron system. Computron's security subsystem performs authentication of a user when the user attempts to start the software. Relying on the authentication performed by the Unix security, Computron validates only that the Unix user account is linked to a Computron user record in the Computron User file (USERFIL3). If such a record exists, the user is allowed access to the system.

If a user record does not exist for the Unix user, the system displays the message: `XXX IS AN INVALID COMPUTRON USER` (where XXX represents the Unix user ID.)

☞ *Note:* Computron's Web clients (WEBdesk) validate against the Unix password file when a user initiates a Computron session from the User Logon window. The only exception is when the user is a Computron guest user. A guest user is a Computron user that logs on to the WEBdesk client using a logon ID and Password that are not valid on the Unix system. See Section 1.2, "User File Maintenance," for additional information.

RDBMS Authentication

After gaining access to the Computron application, a user must pass through the RDBMS security system in order to connect to the relational database that stores Computron data. Computron's User file contains information that enables the Computron user to log onto the relational database. As discussed in Section 1.2, "User File Maintenance," the information entered in the DB ID and Password fields for a User file record causes the system to handle the relational database logon differently. Regardless of the method used to log a user onto the relational database, the following events occur:

- ◆ The user is running Computron software and performs a function that requires data from the relational database.



- ◆ At this time, the information in the Computron User file is used to log the user onto the relational database.



- ◆ The RDBMS security is invoked to perform authentication of the user.



- ◆ Once logged onto the relational database, processing continues.



- ◆ The user remains logged onto the relational database until the database map is changed or the user exits the Computron session.

Since each RDBMS handles authentication of users differently, a brief description for each RDBMS is provided below:

- ◆ Sybase requires you to create user accounts with user IDs and corresponding passwords.
- ◆ Oracle allows you to set up users within the database using unique IDs and passwords or to create user accounts that share the ID (and optionally, the password) with the Unix system security.
- ◆ MSSQL requires you to create user accounts with user IDs and corresponding passwords.

☞ *Important!* Relational database installations require a special Computron user record, &&A, for every owner of Computron tables. This record is used to connect to the database when creating Computron tables or their stored procedures. Without it, various errors can occur within the Computron applications. For a detailed explanation on adding the &&A user record, see the “Special Database User ID Records,” instructions in Section 1.2, “User File Maintenance,” earlier in this guide.

User Privileges

Just as Unix security and RDBMS security systems extend beyond authentication, Computron’s Security Subsystem allows you to establish an additional layer of protection for Computron programs and data. While the Unix system security and the RDBMS security establish user access privileges at the table level (RDBMS) and file level (Unix), Computron’s Security Subsystem controls access to the programs and data used by the Computron applications. The Security Profile allows the Computron administrator to establish access privileges at the Computron record level for those records created exclusively for Computron programs on an application-by-application basis.

Computron's security subsystem is comprised of four major components: the User file, user security profiles, menus and screen images.

- ◆ Computron's User File (USERFIL3) – The User file contains one record for each user permitted access to the Computron applications. The Computron user ID is established in the User file. Additionally, parameters such as the default Database Map file and location, operator level, security level, user group, user menu and company array are established and assigned to the user in the User file.
- ◆ Computron User Security Profiles – User security profile records enable you to apply an additional level of security to each Computron financial application, independent of other Computron financial applications. Ownership codes and record classes are established via the user's security profile. The company access parameters assigned to the user in the User file can be overridden by the profile record.
 - ☞ *Note:* Operator level is assigned to a user in both the User file and the user security profile. When profiles are implemented for the given application, the lower operator level always takes precedence.
- ◆ Computron Menus – Computron Menu files contain a definition for each selection accessible from the menu. Format parameters, program parameters and security parameters are specified for each menu selection. Security parameters assigned to each menu item include security level, allowed user or user group lists, passwords and Database Map files.
- ◆ Computron Screen Image (SCNIMAGE) – Computron screen image files contain the display information for Computron program windows. Screen images can be universal for all users or can be developed to display for a particular user or user group. Additionally, security level parameters can be assigned for each data entry field and screen option (buttons or function keys, depending on the presentation) available.

The above security subsystem components are maintained via the following Computron utilities:

Elements of Security Subsystem	Maintenance Utility
Computron's User File	User File Maintenance
Computron User Security Profiles	User Profile Maintenance
Computron Menus	Menu File Maintenance
Computron Screen Images	Soft Screens Maintenance

The security subsystem enables you to establish various levels of security within the Computron applications. The security is applied to four different types of access: function access, company access, screen access and data access.

- ◆ Function Access – Access to specific Computron functions is determined by a combination of Menu file security information and corresponding information contained in the User file. Users are restricted or allowed access to a menu item based on:
 - The security level applied to each menu item as compared to the user's security level.

- A list of users (or user groups) allowed or disallowed access to each menu item as compared to the user's Computron ID or the Computron group assigned to the user in the User file.

Additionally, users are limited to the menus and submenus they can access by the initial menu that displays for them as defined in the User file.

- ◆ Company Access – Entries in the User file and the user's security profile control the user's ability to access data related to particular companies.
- ◆ Window Access – Screen options (buttons or function keys, depending on the presentation) and individual fields on Computron windows can be secured through a combination of operator level and security level in the user's User file record and the user's security profile. For example, you can restrict a user from deleting records in a maintenance utility by inhibiting the delete option for that window. This type of security is only available in conjunction with Computron's Soft Technology module.
- ◆ Data/Record Access – A user's access to data is controlled by several factors:
 - The user's Database Map file and the ability to change the Database Map file in use, as determined by the user's record in the User file.
 - The tables a user can access and the types of operations the user can perform on those tables as controlled by the RDBMS security.
 - The files a user can access and the extent of the access rights as determined by the Unix system security.
 - Specific records within tables or files based on the company number, ownership code or record class associated with the record as implemented through Computron maintenance utilities.

☞ *Note:* For more detailed information on establishing file access rights, refer to "Recommended Permissions for Unix," at the end of this section.

The Security Mechanism

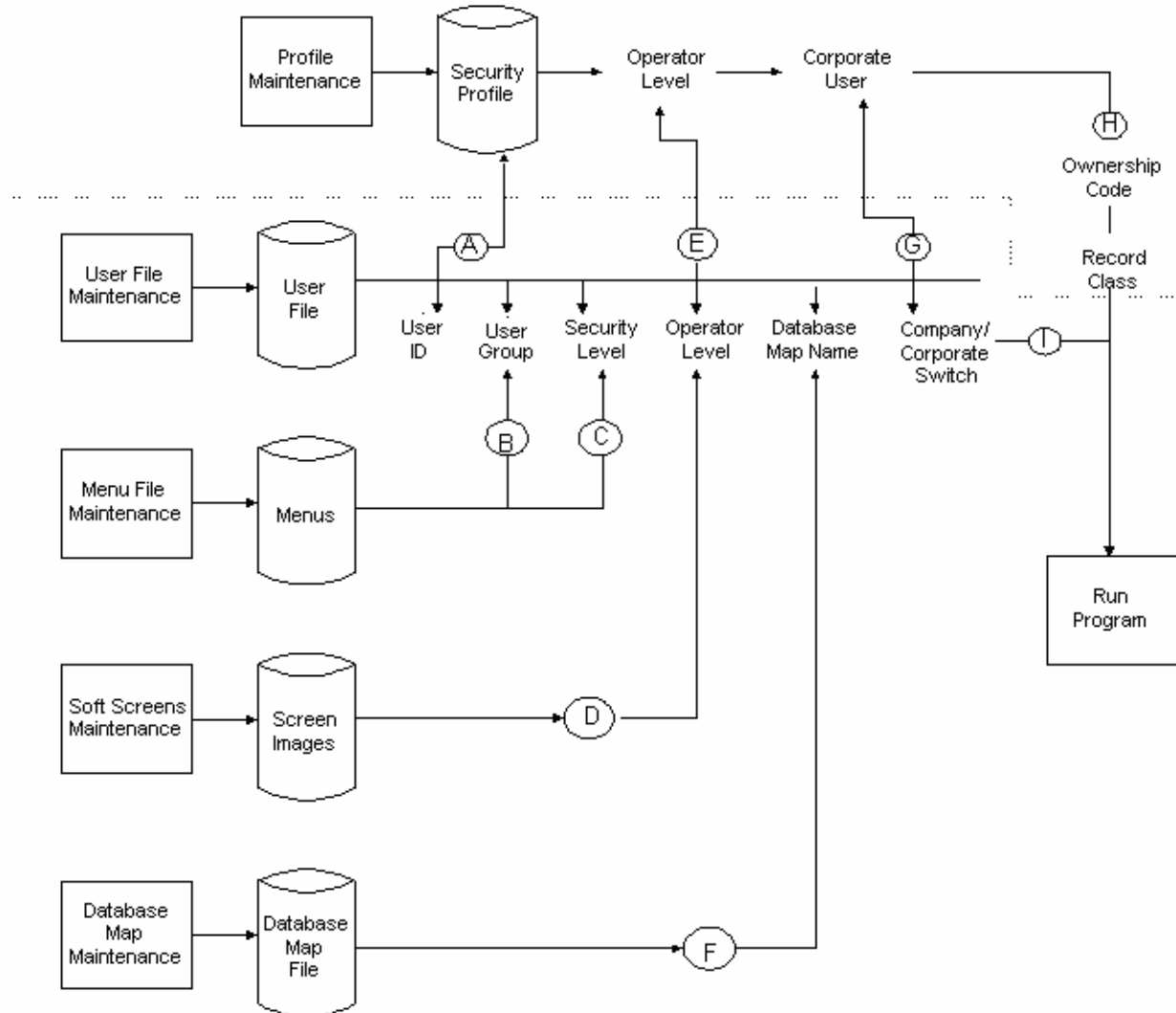
Each element of the security subsystem provides security information; however, the effectiveness of the subsystem is a result of the interdependencies of its components. The relationships between the Computron programs that comprise the security subsystem are illustrated on the following page.

This diagram shows each element of the security subsystem, the security information it stores and how this security information impacts the other elements of the subsystem.

- A. Profile Security is optional for each Computron application (e.g., Accounts Payable, General Ledger, etc.). Security profile records are created for each user in the User file that can access the application. When profile security is implemented, Computron applications access the user's security profile using the Computron user ID as defined in the User file. Additionally, a default security profile record (@@@) can be established. If profile security is in use and there is no security profile record for the Computron user ID, the system uses the security options established for the default profile record, @@@ if that record has been created.

- B. A user is associated with a Computron user group. Security applied to the Menu file includes a list of users or user groups who are allowed or disallowed access to each menu selection.
- C. A user's security level (SL) is established in the User file. This parameter is compared to the security level assigned to each menu item defined in the Menu file to determine whether the user is allowed access to each menu function.


Fig. 2.1-1
Security Subsystem Components



- D. A user's operator level (OL) is used in conjunction with parameters set in the screen images (through Computron's Forms Editor for the Visual Basic programming system or Soft Screens Maintenance) to control a user's access to function buttons and data entry fields on program windows.

- E. The operator level (OL) parameter is established in both the User file and the user's security profile. The lower of the two takes precedence. It is suggested that the highest operator level for the user be established in the User file, allowing the individual security profile entries (by application) to override and lower the user's effective operator level.
- F. The User file contains the default database map name and location. The system uses this information to locate the data that the user is allowed access.
- G. The company/corporate switch in the User file indicates whether the user can change the current database map in use. This switch determines whether the Change Database Map button appears on the toolbar in the WEBdesk presentation and whether the user can access the Change Database Map window via PF-29 on a menu screen in character cell presentations.
- H. This switch in the User file, as well as the Corporate User field in the security profile, also determines whether the user can access consolidated information i.e., information which spans multiple companies.
- I. Ownership code ranges and record class access are specified only in the security profile and apply to the Computron application for which the security profile is established. Access to individual records can be allowed, disallowed or restricted in some way based upon these parameters.
- J. A list of companies to which the user is allowed or disallowed access is denoted in both the User file and the security profile.

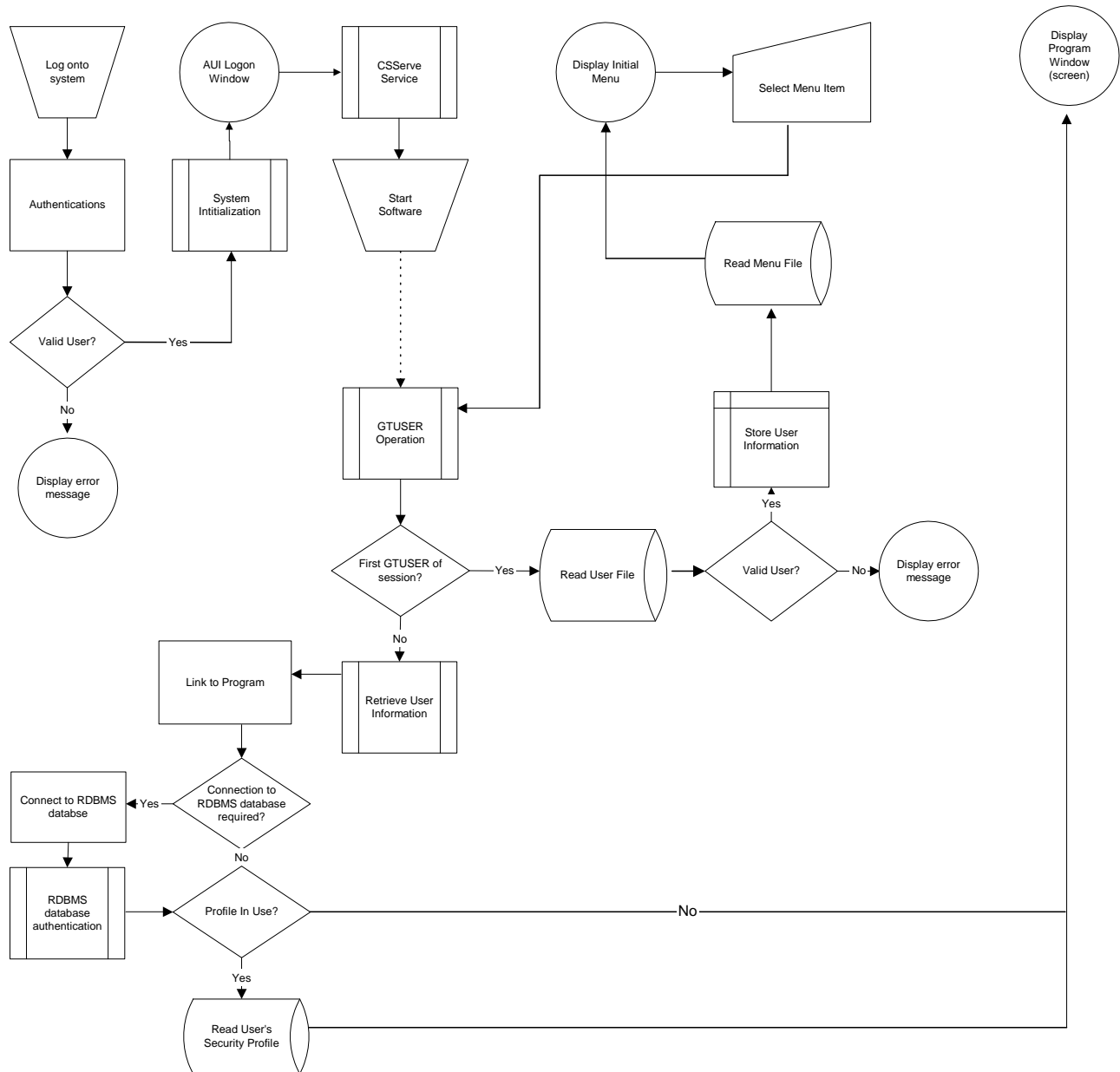
The impact of the various components of the security subsystem on the user's actions that begin with logging onto the system are illustrated in the Security Subsystem Processing flowchart later in this section. This flow chart denotes each step performed by the user and each step consequently performed by the system.

 *Note:* All system tasks are not specifically enumerated.

User Logs onto the Unix System

1. The Unix security validates the user based on the Unix current user account.
 - If the user does not enter a valid password, Unix security does not grant the user access to the system.
 - If the user enters a valid user account and corresponding password, the system allows the user to continue.
2. The Unix security system initializes the user's session based on the user account ID and password:
 - assigns directory and file access privileges to the user.
 - passes the Computron logon request to the CSServe connection service.
 - may also execute an initial program. (If a Computron program is designated, that program will handle additional levels of security assignment.)

**Fig. 2.1-2
Security Subsystem Processing Flow**



User Starts Computron Software

1. The CSServe connection daemon verifies the user logon ID with the Windows security subsystem. If the logon information is correct, then CSServe launches a new process on behalf of that user:
 - ◆ For AUI clients, CSServe daemon launches the runtime WRUNSQL executable and the CSHSDRVR program.

- ◆ For Telnet clients (character cell), CSServe launches the runtime WRUNSQL executable and the MENUPROG program.
- 2. The first time the menu display program is run during a logon session, the system reads the User file (USERFIL3), checking for a record that links the Unix user account to a Computron user ID.
 - If the ID is not listed, the following message displays:

```
Invalid User ID or Password. Please try again.
```
 - If the ID is listed, the system continues.
- 3. The system performs an operation known as a GTUSER. This operation essentially stores and retrieves all the security information from the User file (with database passwords encrypted) in a shared memory area that can be referred to at any time by any Computron program.
 - The system determines if this is the first GTUSER operation for this session.
- 4. For Telnet clients, the system reads the initial Menu file.
 - By default, the initial Menu file is the file, START, located in the library, CIUVMENU, on the volume EXEC.
 - If a user menu is specified in the User file record of the logged-on user, that menu overrides the default menu, START.
 - The system displays the initial menu.

☞ *Note:* When interacting with the Menu file, the security level field (SL) from the User file controls whether the user has access to menu selections. The security level can be set to any number from 0–9. If a menu selection requires a security level higher than that assigned to the user, the menu selection is not displayed on the user’s menu.
- 5. For any Computron client, the Desktop reads the selected application menu tree (e.g., the main General Ledger menu tree) currently cached on the local PC hard drive/Web server. This tree is then presented to the user. The menu in the cache directory is automatically updated whenever the Computron client confirms that the menu on the server is different than the corresponding one in the PC/Web server cache directory.

User Selects Menu Option

1. After merging the menu selection with GTUSER information, the system determines the Database Map file to use for the selected menu option. The database map information stored by the initial GTUSER operation is used unless the selected menu option contains a database map override.
2. The system invokes the appropriate program or procedure.
3. Information is passed to the application in a GTUSER record that is stored in shared memory.
4. The system determines whether the selected program or procedure requires access to the relational database.

- If access is required, information retrieved by the GTUSER operation is used to log the user onto the relational database.
 - If access to the relational database is not required, the system continues.
5. The system determines whether profile security is used by the selected program or procedure.
- If profile security is in use, the system reads the user's security profile and updates security information accordingly. With the exception of the user's operator level, any fields that exist in the profile, that also exist in the User file, override the User file values. The lower of the two operator levels is always used. Whenever a program accesses a record, it checks the Ownership Code and Record Class fields.
 - If profile security is not in use, the system continues.
- ☞ *Note:* The ownership codes entered in the security profile are used in conjunction with the ownership codes assigned to any one of Computron's code maintenance or transaction entry programs. These programs allow the user to enter a four-digit code for each record maintained on the particular system. The code maintenance programs on Computron's financial systems also allow the user to assign a record class to each record. The level of record class access (i.e., whether the user can display, print, copy, modify or delete the record) is defined for each user in the security profile.
6. The system reads the screen images.
- The system determines whether any security restrictions are imposed on the user by the screen image. The screen image file (SCNIMAGE), database files or client-side programs can contain security level indications, allowing and disallowing access to buttons or function keys and/or field entry.
 - The system displays the program window to the user.

Special Processing Notes

1. The initial GTUSER operation, which reads the User file, is performed the first time the menu subsystem (MENUPROG program under the character cell presentation or the CSHSDRVR under any client presentation) is run during the logon session. Thus, if information contained in the User file is modified, the affected user must log off the Computron system and restart the software before modifications take effect.
2. You can override individual user non-security information by entering GTUSEROV data via a procedure. For example, you may want to override the database map in use. (For more information, refer to the *Soft Screens Users Guide*.)
3. If you attempt to run any Computron program from outside the Computron security environment, the system initiates a GTUSER operation to read the User file and determine whether you have a security level of nine. If your security level is below nine, access to the program is denied.
4. Each time the system invokes a child task, the GTUSER shared memory area is updated. Subsequent tasks use this updated information.

5. If the Print Class (PC) or Print Mode (PM) fields are left blank in the User file, the system uses values set in the service configuration or the system defaults (PC=A and PM=H).

Recommended File and Directory Permissions for Unix

The table below represents the minimum Unix permissions for files and directories contained in a generic Computron Software environment (where `/...` represents the file system in which the Computron application software resides).

☞ *Important!* This listing assumes that all of the Computron users belong to the group `ctron`. File and directory ownership must always be `ctronsys` and `ctron`, unless otherwise noted. Keep in mind, however, that every installation is unique and therefore, may require specific permissions for other users.

Directories/Files	Permissions	Ownership
<code>/...</code>	<code>drwxrwxr-x</code>	<code>ctronsys ctron</code>
<code>/.../misc</code> Files: <code>/.../misc/</code>	<code>drwxrwx---</code> <code>-rwxr-xr-x</code>	<code>ctronsys ctron</code>
<code>/.../profiles</code> Files: <code>/.../profiles/</code>	<code>drwxrwx---</code> <code>-rw-r-----</code>	<code>ctronsys ctron</code>
<code>/.../vbpc</code> Sub Dirs: <code>/.../vbpc/exec</code> Files: <code>/.../vbpc/exec/</code>	<code>drwxrwxr-x</code> <code>drwxrwx---</code> <code>-rw-rw----</code>	<code>ctronsys ctron</code>
<code>/.../data</code> Sub Dirs: <code>/.../data/cixxdemo</code> Files: <code>/.../data/cixxdemo/</code>	<code>drwxrwx---</code> <code>drwxrwx---</code> <code>-rw-rw----</code>	<code>ctronsys ctron</code>
<code>/.../exec</code> Sub Dirs: <code>/.../exec/CIXXCOPW</code> Files: <code>/.../exec/CIXXCOPW/</code>	<code>drwxrwxr-x</code> <code>drwxr-x---</code> <code>-rw-r-----</code>	<code>ctronsys ctron</code>
<code>/.../exec/xxxxctl</code> (ctli and epic directories) Files: <code>/.../exec/xxxxctl/</code>	<code>drwx-w----</code> <code>-rw-r-----</code>	<code>ctronsys ctron</code>
<code>/.../exec/xxxxload</code> directories Files: <code>/.../exec/xxxxload/</code>	<code>drwxrwx---</code> <code>-rw-r-----</code>	<code>ctronsys ctron</code>
<code>/.../exec/xxxxrun</code> directories Files: <code>/.../exec/xxxxrun/</code>	<code>drwxr-x---</code> <code>-rwxr-x---</code>	<code>ctronsys ctron</code>
<code>/.../exec/xxxxmenu</code> directories Files: <code>/.../exec/xxxxmenu/</code>	<code>drwxr-x---</code> <code>-rw-r-----</code>	<code>ctronsys ctron</code>
<code>/.../exec/xxxxsni</code> Sub Dirs: <code>/.../exec/xxxxsni/</code> Files: <code>/.../exec/xxxxsni/vb4/</code>	<code>drwxrwx---</code> <code>drwxrwx---</code> <code>-rw-rw----</code>	<code>ctronsys ctron</code>

Special Directories and Files

By default, these special directories reside in the `/.../exec` directory. However, you can maximize your disk space by storing them in a different location:

Special Directories	Permissions	Ownership
<code>/.../var/spool</code>	<code>drwxrwx---</code>	ctronsys ctron
<code>/.../var/worklibs</code>	<code>drwxrwx---</code>	ctronsys ctron
<code>/.../var/logs</code>	<code>drwxrwx---</code>	ctronsys ctron
Special Files	Permissions	Ownership
<code>/.../exec/csuvlipc</code>	<code>-rw-r-----</code>	ctronsys ctron
ctron Directory	Permissions	Ownership
<code>/.../exec/_ctron_</code> Sub Dir: <code>/.../exec/_ctron_/putparms</code> Files: <code>/.../exec/_ctron_/lockout</code> <code>/.../exec/_ctron_/userfil3</code>	<code>drwxrwx---</code> <code>drwxrwx---</code> <code>-rw-rw-r--</code> <code>-rw-rw-r--</code>	ctronsys ctron

☞ *Note:* All other files and subdirectories under `/.../exec/_ctron_` require `[rwx]` for the owner and `[rx]` for the group.

Computron uses the CSSORT routine to build files during the sort process. While sorting, CSSORT needs temporary disk space equal to twice the size of the sort file. By default, the `TMPDIR` setting in Computron's start script specifies a work library (worklib) for Computron user temporary files. However, if another temporary directory is specified in the `local.start` file (e.g., `TMPDIR=/usr/ctron_tmp`), then permissions must be set to allow Computron users to write to that directory. For example:

Special Directory	Permissions	Ownership
<code>/usr/ctron_tmp</code>	<code>drwxrwx---</code>	ctronsys ctron

☞ *Note:* For more information about Computron's start script and the `local.start` file, refer to section 5.7, "Computron's Start Script."

2.2 Ownership Code Maintenance

Introduction to Ownership Code Maintenance

This function is used to establish ownership codes to be used as a part of the Security subsystem. Ownership codes created through this function can be tied to individual records through application maintenance functions. Once an ownership code is entered for a record, individual users can thereafter be allowed or disallowed access to the record, based on parameters established via the Profile Maintenance function. In the User profile, a user can be allowed or disallowed access to records with specified ownership codes, or within a range of ownership codes. In addition, a default ownership code for the user can be assigned via Profile Maintenance. See Section 2.4, “User Profile Maintenance,” for additional information.

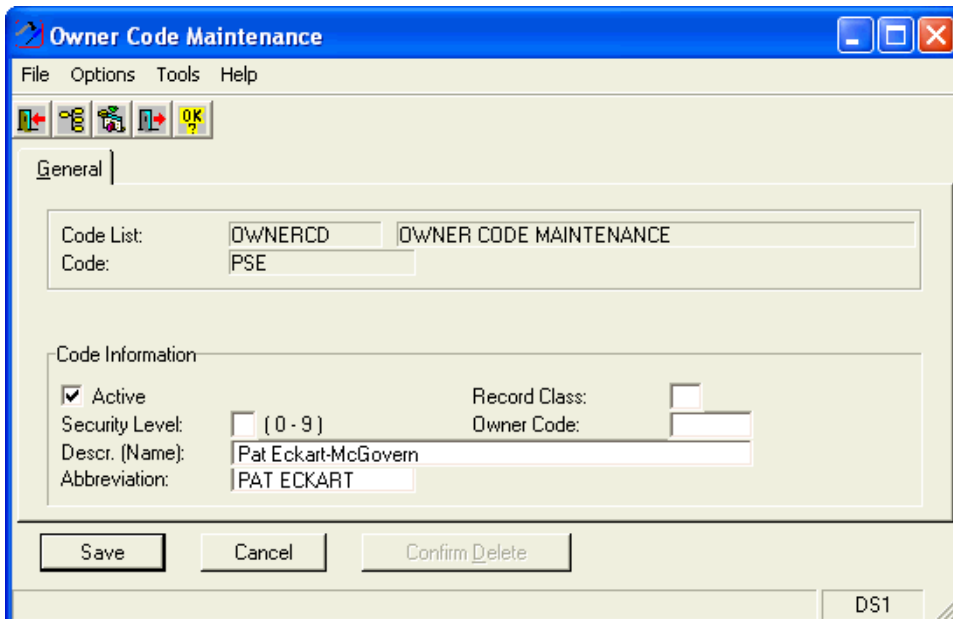
☞ *Note:* On entry windows, the Ownership Code field is always Q-Markable.

Owner Code Maintenance Window (KS1)

The system prompts for entry of an ownership code (Q-Mark). Enter an alphanumeric code up to three characters for the ownership code to be maintained.

Next, click one of the maintenance buttons (Add, Change, Delete or Copy) at the bottom of the window to open the data entry (DS1) window.

Owner Code Maintenance Window (DS1)



Ownership Code Maintenance Window (DS1)

Active

Enter the status of the ownership code. This field appears on Q-^{Mark} windows.

- Indicates the code is active and can be used.
- Indicates the code is inactive.

Default:

Record Class

Enter the record class to assign to this code. This field is used in combination with parameters established in the User profile to restrict access to this record to a specified class of users. Valid values are A–Z.

Security Level

This security field is used in combination with data established in the User file to restrict the use of this code. If entered, this record is available to users with an equal or higher security level. Valid values are 0–9.

Owner Code

This field is contingent upon parameters established in the User profile. Only users with this ownership code can access this record for any purpose.

Descr. (Name)

This is a thirty position alphanumeric description of the ownership code. This field appears on Q-^{Mark} windows.

Abbreviation

This is a ten position alphanumeric description of the ownership code. This field appears on Q-^{Mark} windows.

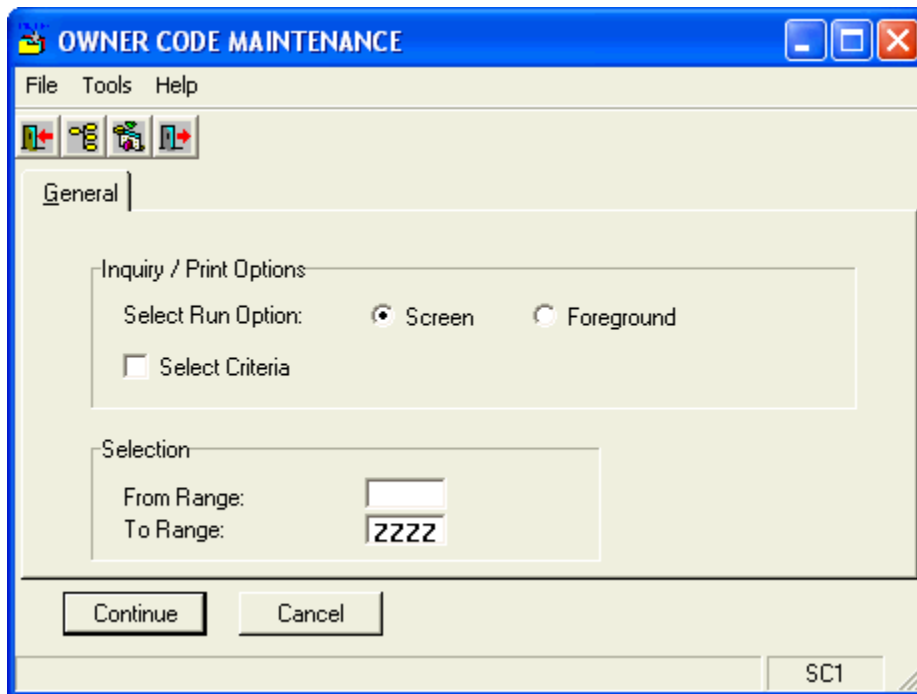
2.3 Ownership Code Inquiry / List

Introduction to Ownership Code Inquiry / List

This function is used to display a listing of the ownership codes established via the Ownership Code Maintenance function. This data can either be displayed online or spooled to a print file.

Owner Code Maintenance Window (SC1)

The following window displays when you select the Ownership Code Inquiry/List function.



Ownership Code Inquiry/List Window (SC1)

Select Run Option

This is the processing option that produces the online display or the printed report. Valid options are:

Screen – This enables you to review a list of the data online, as well as to display detail for each item on the list.

Foreground – This option produces a print file, processing the request in foreground. Use of this option restricts use of this window for other processing.

Default: Foreground.

Select Criteria

This determines whether to use a query window to specify additional selection criteria. Valid options are:

Displays the Select Query Window (SC99), which allows you to further define the selection criteria.

Do not allow modification of the selection criteria.

Default:

From Range

Enter the beginning of the range of codes to include in the inquiry/list.

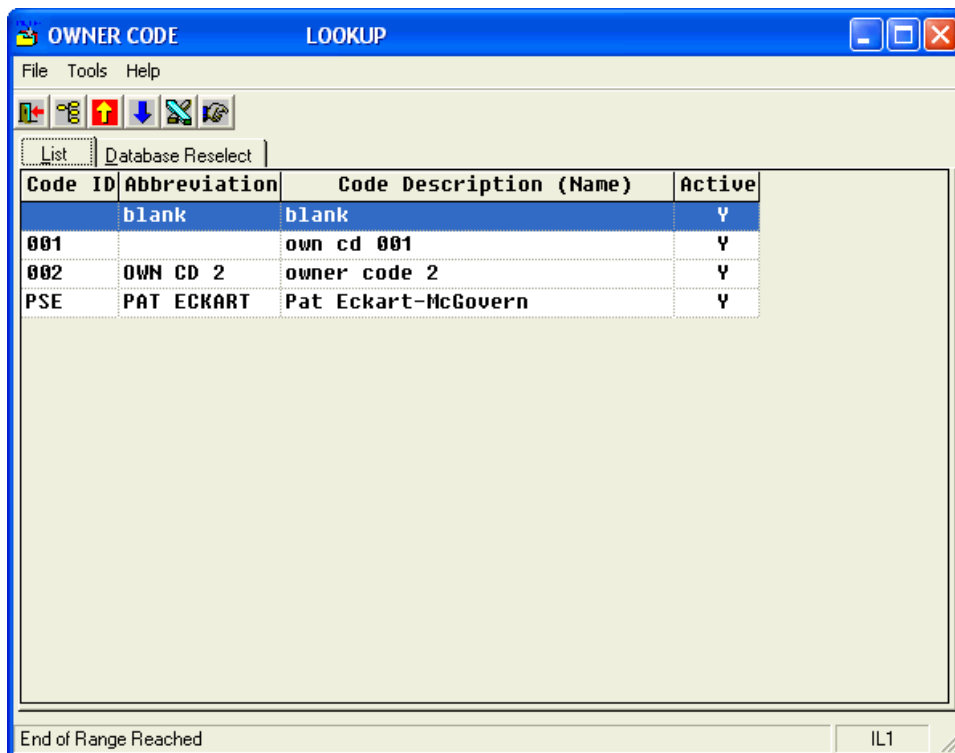
To Range

Enter the end of the range of codes to include in the inquiry/list.

Ownership Code Report Options

Owner Code Lookup Window (IL1)

When the Screen option is selected, an inquiry window displays.



The screenshot shows a window titled "OWNER CODE LOOKUP" with a menu bar (File, Tools, Help) and a toolbar. Below the toolbar are buttons for "List" and "Database Reselect". The main area contains a table with the following data:

Code ID	Abbreviation	Code Description (Name)	Active
	blank	blank	Y
001		own cd 001	Y
002	OWN CD 2	owner code 2	Y
PSE	PAT ECKART	Pat Eckart-McGovern	Y

At the bottom of the window, there is a status bar that reads "End of Range Reached" and a small box containing "IL1".

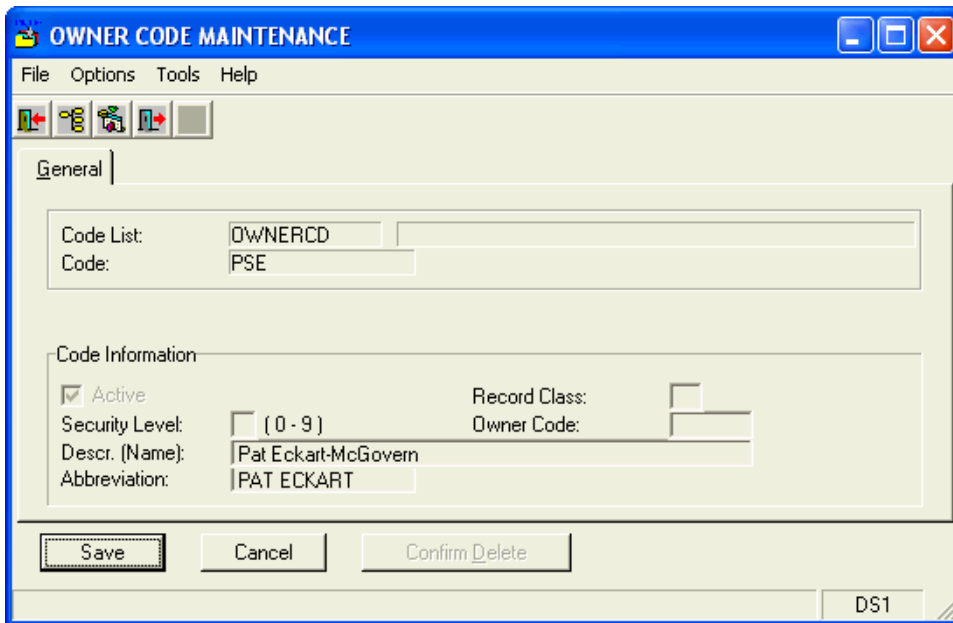
Ownership Code Inquiry Window (IL1)

To display the details of a listed record, you can either:

- ◆ highlight it and click the Choose button.
- ◆ double-click the desired code.

An inquiry detail window displays.

Owner Code Maintenance Window (DS1)



OWNER CODE MAINTENANCE

File Options Tools Help

General

Code List: OWNERCD
Code: PSE

Code Information

Active
Security Level: [] (0 - 9) Record Class: []
Descr. (Name): Pat Eckart-McGovern Owner Code: []
Abbreviation: PAT ECKART

Save Cancel Confirm Delete

DS1

Ownership Code Detail Window (DS1)

Print Options

The following sample report is available for the inquiry/list function.

Ownership Code Listing

```

* Computron Software, LLC ***** Mon Apr 30, 2007 11:02 AM *
*
* --- Owner Code Maintenance ---
*
* ..... Page 4 *

```

Code ID	Abbreviation	Code Description (Name)	Active
-----	blank	blank	Y
001	OWN CD 1	own cd 001	Y
002	OWN CD 2	owner code 2	Y
FIN	FINANCE	Finance Department	Y

2.4 User Profile Maintenance

Introduction to User Profile Maintenance

The User Profile Maintenance function allows you to establish and maintain User Profile records, which complement corresponding records in the User file. One User file record is created for each user who is permitted access to Computron functions. Once a User file record has been established for a particular Computron user, certain security parameters are in effect for any Computron function the user accesses.

User Profile records are also created for individual users. They are, however, used to establish security on a product basis; particularly when the security requirements of one Computron product differ from those requirements already established in the User file record.

For example, suppose a user is the supervisor of the Accounts Payable department. This user needs to have access to most of the functions in the Accounts Payable system and needs to view information in the General Ledger system, but should not be allowed to change information in the General Ledger system. In this scenario you can utilize the User Profile in the Accounts Payable system to establish a high security level for the user in this system while maintaining a lower security level in the General Ledger system via the User file or the General Ledger User Profile. In fact, you can customize security parameters as needed by creating one User Profile for each Computron product a user accesses.

User Profile records contain some of the same fields as User file records. For example, both types of records allow the user to establish a default company for a particular Computron user ID.

Because User Profile records are product specific, they automatically override all corresponding fields in the User file — except for the Operator Level field. Regardless of the operator level values in the User Profile and the User file, the lower of the two always takes precedence. Therefore, when establishing security on the system, it is a good idea to assign the User file the highest operator level that individual is allowed. Thus, you can automatically override it for a product by creating a User Profile record with a lower operator level.

Example: Suppose Computron user ID, CTR, has been assigned an operator level of 9 via User File Maintenance, which allows file maintenance on a system-wide basis. To restrict the user from maintaining files on the General Ledger system, create a User Profile record on the General Ledger system with a lower operator level of 8.

The User Profile offers additional levels of security unavailable in the User file.

Suppose in the Company Parameters Maintenance function you assign an ownership code, AAAA to a General Ledger Company record. To enforce this code, i.e., indicate which users can and cannot access the company record, corresponding entries to the Ownership Security fields in the User Profile must exist. Ownership Security fields are used to list a range of ownership codes and indicate whether the user is allowed or disallowed access to them.

Example: To disallow profile logon ID, CTR, from accessing GL company records that have been assigned an ownership code, AAAA, enter the following values in the Ownership Security section of the Profile Maintenance Window (DS2):

In the User Profile Maintenance function (General Ledger system) for user profile logon ID, CTR, establish the following parameters:

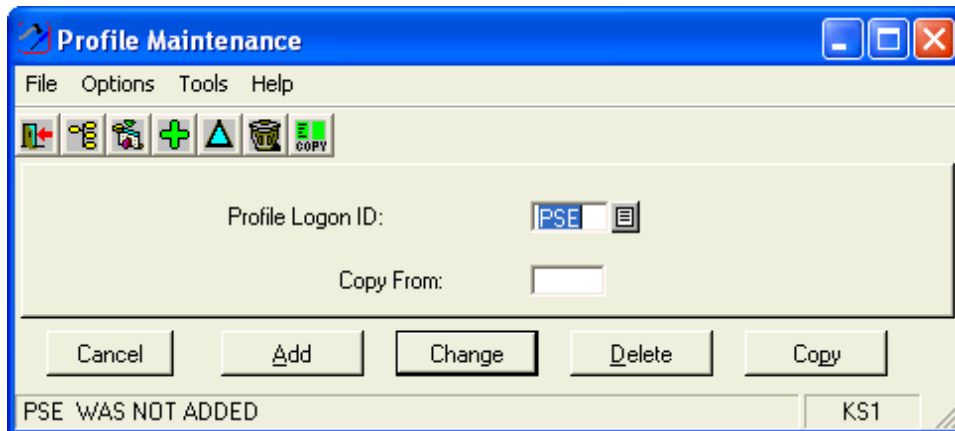
Allow/Disallow	From — To
D	AAAA — AAAA

☞ *Note:* To activate User Profile Security for a particular system:

- ◆ Access the Global Parameters option via Soft Screen Maintenance for the Universal Utilities screen image file. Select (set to Yes) the Check User Profile field on the Global Parameters Window (GL2).
- ◆ Access the Global Parameters option via Soft Screen Maintenance for the system's screen image file (AP, GL, etc.). Select (set to Yes) the Check User Profile field.
- ◆ Access the Global Parameters/Environment Maintenance function for the system (AP, GL, etc.). Select (set to Yes) the Profile In Use (Use Profile Security) field and enter PROFILE in the Profile List Name field if present (Profile Q-Mark Code in TB).

Profile Maintenance Window (KS1)

The User Profile Header Window is the first window to display in User Profile Maintenance. The system prompts for entry of a Profile Logon ID (Q-Mark). Choose a maintenance option (Add, Change, Delete, or Copy) by clicking the appropriate button.



User Profile Header Window (KS1)

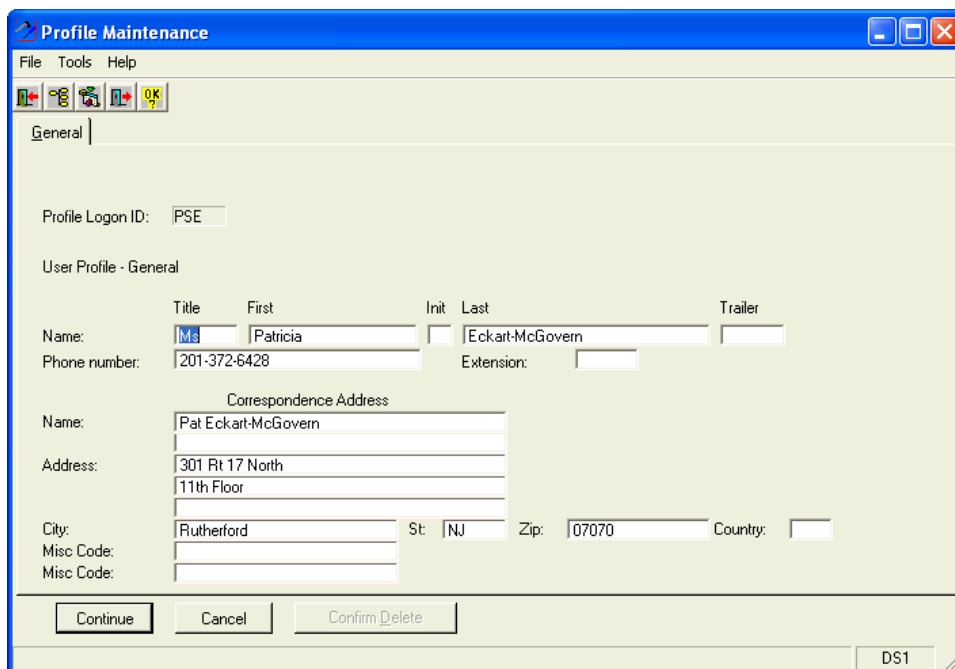
Profile Logon ID (Q-Mark)

This is the Computron user ID of the record to maintain.

- ☞ *Note:* One special record is required in the User Profile file, @@@. If @@@ is not available and a user without a profile logon ID logs on, then the program cancels. The @@@ record is used to set default security levels for any Computron user ID which has not be explicitly entered in the file. Thus, the security parameters entered for the @@@ record is ordinarily the lowest level of access allowed to any user in the product.

Table 2.4-1 Computron Options — User Profile Header Window	
Button	Function
Cancel	Exit User Profile Maintenance.
Add	Add a new user profile record. This displays the User Profile Detail Window (DS1) where you can add the user profile record.
Delete	Delete a User Profile record.
Change	Change an existing User Profile record. This displays the User Profile Detail Window (DS1) where you can make changes to the User Profile record.
Copy	Copy an existing User Profile record.

Profile Maintenance Window (DS1)



User Profile Detail Window (DS1)

Name

Enter the name of the Computron user. You enter data for five fields: Title (e.g., Mr., Ms., Dr., etc.) First, Init, Last, Trailer (e.g., Jr.).

Note: The user is required to enter a value in the Last field.

Phone number

Enter the phone number of the Computron user. This is not a required field.

Extension

Enter the phone number extension of the Computron user. This is not a required field.

Correspondence Address

Enter the correspondence address of the Computron user. It is comprised of six fields: Name, Address, City, St, Zip and Country. The first address line is a required field, though the remaining lines are optional.

Misc Code

This field is not currently implemented.

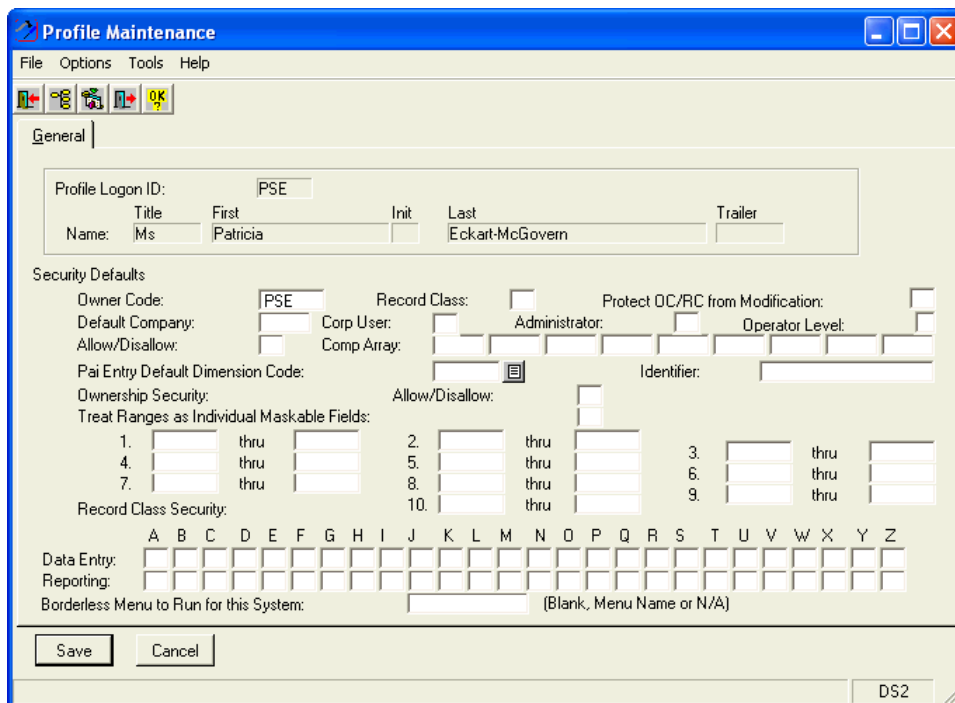
Misc Code

This field is not currently implemented.

The Profile Maintenance Window (DS2) displays when the Continue button is clicked.

Profile Maintenance Window (DS2)

The default information that displays on the window is extracted from the corresponding User file record for the profile logon ID. For example, the default company that displays on this window is extracted from the corresponding field in the User file.



The screenshot shows the 'Profile Maintenance' window with the 'General' tab selected. The 'Profile Logon ID' field contains 'PSE'. The 'Name' section includes 'Title' (Ms), 'First' (Patricia), 'Init' (blank), 'Last' (Eckart-McGovern), and 'Trailer' (blank). The 'Security Defaults' section includes 'Owner Code' (PSE), 'Record Class' (blank), 'Protect OC/RC from Modification' (checkbox), 'Default Company' (blank), 'Corp User' (checkbox), 'Administrator' (checkbox), 'Operator Level' (checkbox), 'Allow/Disallow' (checkbox), and 'Comp Array' (checkbox). The 'Pai Entry Default Dimension Code' is blank, and the 'Identifier' is blank. The 'Ownership Security' section includes 'Treat Ranges as Individual Maskable Fields' (checkbox) and a grid of 10 'Allow/Disallow' checkboxes. The 'Record Class Security' section includes a grid of checkboxes for letters A through Z. The 'Data Entry' and 'Reporting' sections include grids of checkboxes. The 'Borderless Menu to Run for this System' field is blank. The window has 'Save' and 'Cancel' buttons at the bottom left and 'DS2' at the bottom right.


User Profile Detail Window (DS2)

Ownership Code

This field establishes the ownership code to use as a default value in any record maintained by the Computron user.

Record Class

This field establishes the record class to use as a default value in any record maintained by the Computron user. Valid entries are any letter from A–Z.

 *Warning! Use caution when setting the ownership code and/or record class fields in the Security Defaults section. A common mistake is setting default values which conflict with the ownership code and/or record class field settings in the following section.*

Protect OC/RC from Modification

This field indicates whether the default ownership code and/or record class assigned to a record can be modified by the Computron user. Valid entries are:

Y – The default value cannot be modified once assigned.

N – The default value can be modified.

Default Company

This field establishes the default value to display in the Company Number field wherever prompted in the product. If a default company has been entered for the user via User file Maintenance, it displays as the initial value in this field.

Validation: The company number you enter here must have been established as valid in the Company file and the Company Parameters Maintenance function for the product.

Corp User

This field indicates whether the user is a corporate user and therefore allowed to run corporate reports. Valid entries are:

Y – The user is permitted to run corporate reports (reports spanning multiple companies).

N – The user is not permitted to run corporate reports. The user is permitted to run reports for individual companies only. When this option is in effect, it is usually coupled with the Allow/Disallow option.

Default: The default value that displays in this field is extracted from the corporate flag (Corp field) in the corresponding User file record. If the Corp field contains a Y or U, this field is updated with a Y; otherwise, it is updated with an N.

Administrator

This assigns system administrator privileges to the user within the product. Valid entries are:

Y – The user has unlimited access to system records, regardless of the value entered in the Ownership Code and/or Record Class fields.

N – The user has limited access to system records. The user's access to system records is dependent on the values entered in the Ownership Code and Record Class fields.

Operator Level

This controls a user's access to operations within programs. This field, which can contain any digit from 0–9, indicates the level of access that the user has to specific buttons and individual fields on windows. A button or field is inaccessible to the user if its operator level is greater than the user's operator level. In fact, any field inaccessible to the user is suppressed from displaying on the user's window.

The system compares this operator level number to the corresponding number in the User file and accepts the lower of the two. Because of this, it is recommended that you assign the User file the highest number so that any lower-level numbers entered in the User Profile automatically take precedence.

If an operator level was entered in the corresponding User file record, it appears in this field as a default.

Operator Levels for buttons and fields are entered via the optional Soft Technology module.

Example: To restrict user, CTR from deleting records in the Journal Maintenance function, assign the Delete button within this function an operator level of eight via the Soft Technology module. Assign an operator level of six to the User Profile record (CTR) via User Profile Maintenance for the General Ledger product.

Allow/Disallow

This field is used if N is the value entered in the Corp User field above. The Allow/Disallow field, when used in conjunction with the Comp Array field that follows, designates which company records the user is able to access. Valid entries are:

A – The user is allowed to access the records of those companies entered in the Comp Array field.

D – The user is not allowed to access the records of those companies entered in the Comp Array field.

Comp Array

This field is used in conjunction with the Allow/Disallow field, this field indicates which company records are available to the Computron user. Up to nine companies can be specified.

PAI Entry Default Dimension Code (Q-Mark)

The value in this field specifies a default dimension for Universal Line Method (ULM) windows.

Identifier

The value in this field specifies a default identifier for ULM windows.

The Ownership Security fields that follow are used in conjunction with the ownership security codes that the user can enter in any Computron file maintenance or transaction entry function. These functions allow the user to enter a four-digit code for each record to maintain within the product. Other users can then be allowed or disallowed access to specific records based on these ownership codes.

Allow/Disallow

This field is used in conjunction with the subsequent From — To field. Valid entries are:

A – The user is allowed access to those records within the range of ownership codes listed in the From — To field.

D – The user is not allowed access to those records within the range of ownership codes listed in the From — To field.

From—To

This field is used in conjunction with the Allow/Disallow field and indicates the range of ownership codes accessible to the Computron user. Up to ten different ranges can be specified.

☞ *Note:* If these fields are blank, users are able to access all records regardless of the value in the Allow/Disallow field.

Example#1: To restrict user CTR from maintaining or printing General Ledger posting accounts that pertain to Department 101, enter the following parameters:

In the Identifier Maintenance function for the department dimension:

Identifier	Ownership Security Code
100	AAAA

On the User Profile Detail window for profile logon ID, CTR:

Allow/Disallow	From — To
D	AAAA — AAAA

Example#2: To allow user CTR to print only those line sets that pertain to posting accounts 1000-2000, i.e., line sets 100-300, enter the following parameters:

In the Line Set Maintenance function:

Line Set Code	Ownership Security Code
100	DDDD
200	EEEE
300	FFFF

On the User Profile Detail Window for profile logon ID, CTR:

Allow/Disallow	From — To
A	DDDD — FFFF

Record Class Security

This option allows you to establish the Computron user's level of access to Computron maintenance and inquiry records.

The file maintenance functions on Computron's financial systems allow the user to assign a record class for each record maintained (a letter ranging from A to Z). User Profile

Maintenance, in turn, allows you to specify the record classes each profile logon ID has access to, in addition to stipulating the level of access granted (read, write, or execute).

Record class security comprises two fields. Different levels of access can be assigned to individual Computron users based on the functions from which the record is being accessed:

Data Entry – The level of access to allow from any data entry (maintenance) function.

Reporting – The level of access to allow from any reporting (inquiry) function.

To allow access to a particular record class, e.g., record class — A for a particular profile logon ID, enter one of the following values (E, R, W, Blank) in the Data Entry and/or Reporting field for the appropriate record class.

E – Allows the user to display or print the record. This option is only valid for Reporting record class assignments.

R – Allows the user to display or print the record (in inquiry functions), and copy the record (in maintenance functions), but does not allow the user to modify the record.

W – Allow the user unlimited access to the record. The user can display, print, copy, modify, or delete the record.

☞ *Note:* For Reporting, W and R have the same effect.

Blank – The user cannot access the record for any purpose.

Record class security can be used in conjunction with ownership security to allow two users different access levels to the same records. By specifying the level of access via record class security, one user can view and modify the record, while the other user is only able to view and copy the record.

Example: To allow profile logon ID, CS1, to view, print and modify a particular range of records and allow profile Logon ID, CS2, to view, print and copy the same record without modification permission:

For profile logon IDs, CS1 and CS2; enter the following parameters:

Ownership Code	Allow/Disallow	From — To	Record Class
AAAA	A	AAAA — AAAA	A

For Profile Logon ID, CS1; enter the following parameters:

Data Entry	Reporting
W	

For Profile Logon ID, CS1; enter the following parameters:

Data Entry	Reporting
------------	-----------

R

☞ *Note:* The Administrator field must be set to N for profile logon ID, CS2. Otherwise the Computron user identified by profile logon ID, CS2 has unlimited access to every record, regardless of any value entered in record class.

Borderless Menu to Run for this System

This field is used by Computron's character cell presentation.

This specifies the menu to use in order to run Borderless Processing for the Computron user. Valid entries are:

Blank – Display the standard Computron Borderless Processing menu when the user selects this option via Function Key PF-30. The standard menu name is QANYMENU. This menu should be included in the standard menu library for the product, e.g., CIAPMENU, CIGLMENU, etc.

Menu Name – Customized user menu name. This menu should be included in the standard menu library for the product, e.g., CIAPMENU, CIGLMENU, etc.

N/A – Borderless Processing cannot be used.

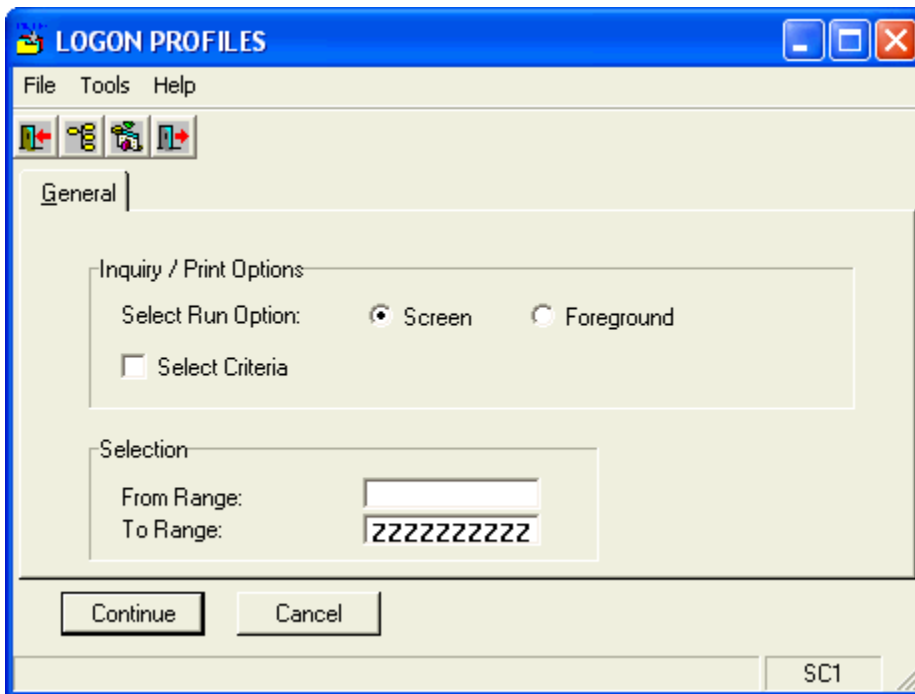
2.5 User Profile Inquiry / List

Introduction to User Profile Inquiry / List

This function displays a listing of the User Profile data established via the User Profile Maintenance function. This data can either be displayed online or spooled to a print file.

Logon Profiles Window (SC1)

The SC1 window displays when you select the User Profile Inquiry/List function.



Logon Profiles Window (SC1)

Select Run Option

These are the processing options that produce online and printed reports. Valid options are:

Screen – Enables you to review a list of the data online, as well as to display detail for each item on the list.

Foreground – Produces a print file, processing the request in foreground. Use of this option restricts use of this window for other processing.

Default: Foreground.

Select Criteria

This field determines whether to use a query window to specify additional selection criteria. Valid options are:

Displays the Select Query Window (SC99), which allows you to further define the selection criteria.

Do not allow modification of the selection criteria.

Default:

From Range

Enter the beginning of the range of User Profile records to include in the inquiry/list.

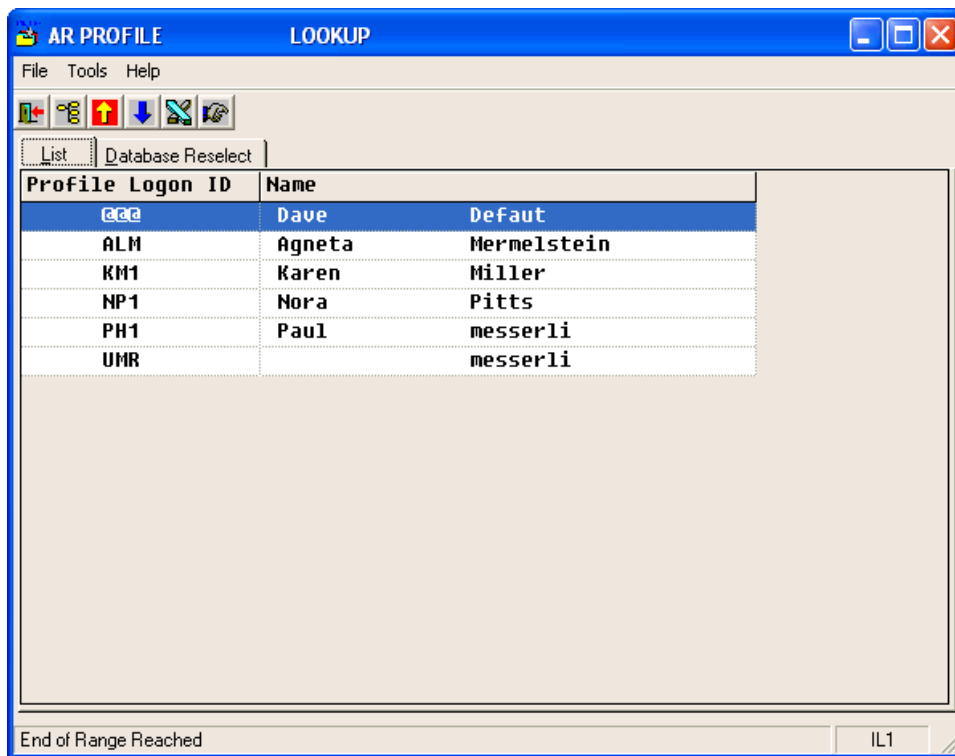
To Range

Enter the end of the range of User Profile records to include in the inquiry/list.

User Profiles Report Options

Screen Option

When the Screen option is selected, an inquiry window displays:



The screenshot shows a window titled "AR PROFILE LOOKUP" with a menu bar (File, Tools, Help) and a toolbar. Below the toolbar are buttons for "List" and "Database Reselect". The main area contains a table with the following data:

Profile	Logon ID	Name	Default
ALM		Agneta	Mermelstein
KM1		Karen	Miller
NP1		Nora	Pitts
PH1		Paul	messerli
UMR			messerli

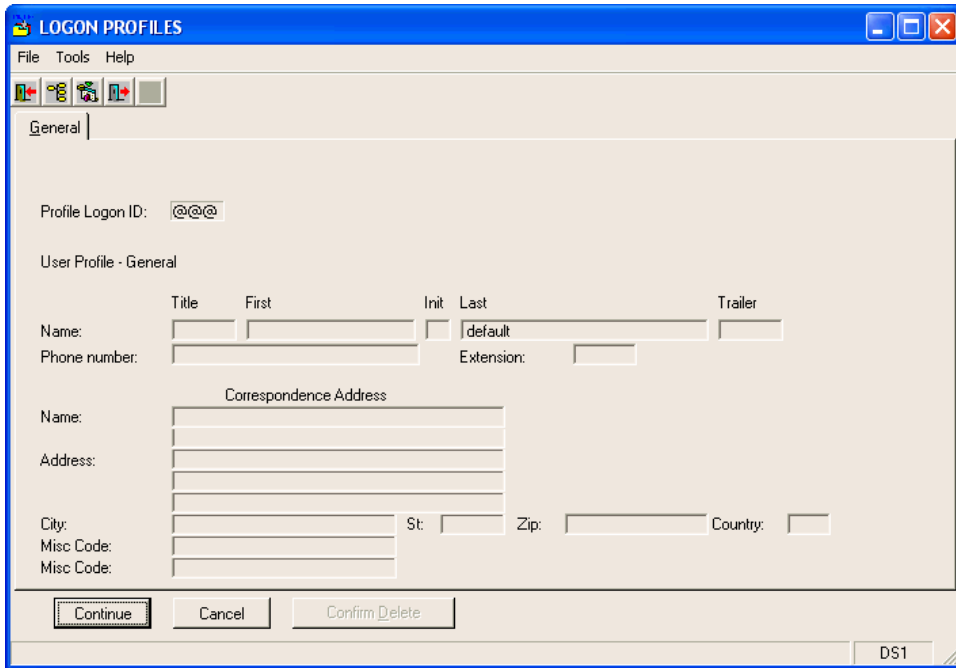
At the bottom of the window, there is a status bar that says "End of Range Reached" and a small box containing "IL1".

User Profile Inquiry/List Window (IL1)

To display the details of a listed code, you can either:

- ◆ highlight it and click the Choose button.
- ◆ double-click the desired code.

An inquiry detail window displays:



User Profile Inquiry/List Detail Window (DS1)

Print Options

The following sample report is available for the inquiry/list function:

User Profile Listing

```

* Computron Software, LLC ***** Mon Apr 30, 2007 1:09 PM *
*                                     --- Profile Maintenance --- *
*                                                                 *
*----- Page 1 *

```

Profile Logon ID	Name		
@@@		default	
@@A		@@A	
BB1	WILLIAM	J BROCKETT	1234
CS	Carol	Selzer	
GL		long	
TM2		monahan	

Chapter 3

Job Processing

3.1 Job Processing – An Overview

Introduction to Job Processing

This section explains the purpose of Job Processing (a.k.a. Batch Stream Processing, BSP), explores Job Processing functionality and defines the steps required to implement Job Processing. Subsequent sections discuss in detail the functions that comprise the Job Processing module.

What is Job Processing?

Job Processing is a special module designed to meet the following common data processing needs:

- ◆ Run a series of jobs in a predetermined sequence with built-in dependencies without supervision or any direct knowledge of the jobs to run.
- ◆ Automatically supply parameters such as dates (as stored in the Job Scheduler Directory).
- ◆ Control how certain functions are run: on an as-needed basis, only as part of a job or conditionally (as based on the success or failure of preceding processes — via return codes).
- ◆ Provide run-time defaults for the initiation of interactive jobs (based on the defaults stored in the Job Scheduler Directory for Job Processor-Prepared Computron programs).
- ◆ Automatically produce multiple copies of reports and easily distribute them to the desired persons and departments.

Job Processing can be used to run any of the following job types:

- ◆ Job Processor-Prepared Computron Applications — These jobs are customized to accept a predefined set of parameters at run time. The parameters are specified once and thereafter used automatically by the system each time the associated job is run. When such jobs are initiated from the Job Processor, user prompts do not appear because the necessary information has already been provided. In addition, variable information such as dates are parameterized and calculated automatically by the system for these jobs, based on the processing date you supply on initiating the Job Processor (e.g., calculate end of month, end of year, etc.). These jobs commonly generate reports.
- ◆ Other Computron Application Processes — Any Computron job can be initiated *interactively* (i.e., with user prompts) from the Job Processor, including both Job Processor-Prepared and non-Job Processor-Prepared Computron application processes. In this case, all user prompts appear as usual.
- ◆ Non-Computron Application Processes and Utilities — Any non-Computron job or script can be run from the Job Processor provided the foreign entities comply with Computron standards for naming and location.

Job Processing can also be used in conjunction with multiple databases. You can direct the Job Processor to run one series (“stream”) of jobs for one database and then shift automatically to another database to run another stream of jobs; return and continue with the next job on the main directory, or initiate the job process for another database.

How Job Processing Works

The following flowchart illustrates how the various components of the Job Processing module interact with one another.

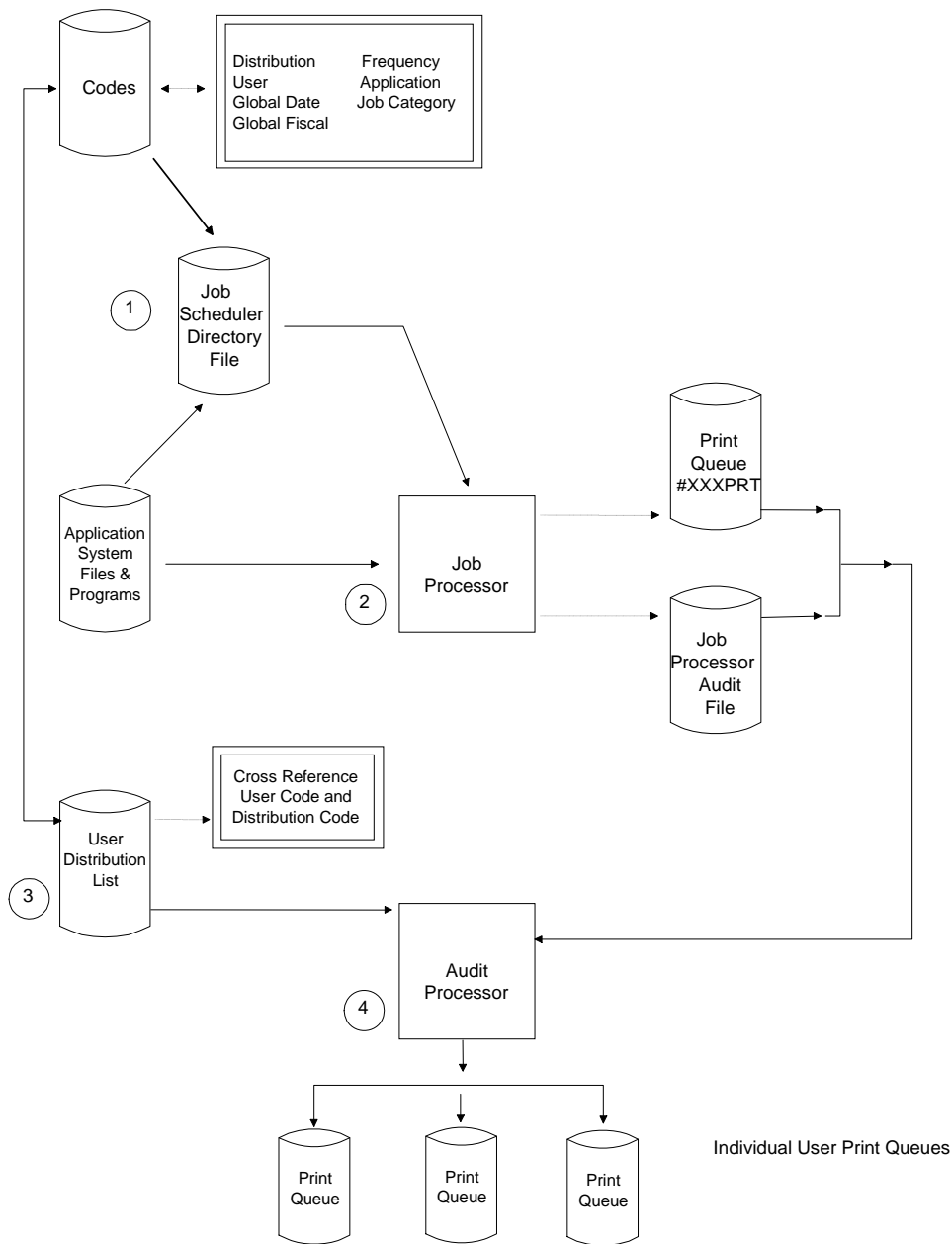


Fig 3.1-1

The Job Processing module has four major components:

- ◆ Job Scheduler Directory
- ◆ Job Processor
- ◆ User Distribution List
- ◆ Audit Processor

The role of these components and how they interrelate is discussed next.

Job Scheduler Directory

The Job Scheduler Directory is maintained either via the Job Scheduler Directory Maintenance function (for Job Processor-Prepared Computron functions) or from a Computron menu selection (for Computron and non-Computron functions). The Job Scheduler Directory contains the information, used by the Job Processor to control the running of each job in the job stream. For each job, it contains information such as:

- ◆ **Sequence Number** — Controls the order in which the jobs are run (as part of the job stream).
- ◆ **Distribution Code** — This code is used by the Audit Processor to determine which users receive copies of reports.
- ◆ **Frequency Code** — Generic frequency codes (system options) offer considerable job scheduling flexibility. For instance, a generic frequency code can indicate that a job is to run every Friday, the last day of each month, the fifteenth of each month or the second Tuesday of each month.

Validation: The Job Processor compares the generic frequency code associated with the job to run to the processing date entered at run time to determine whether or not the job should run.

You are not confined to using the generic frequency codes. You can define additional frequency codes, such as *D* for daily, *M* for monthly and so on. These user-defined codes are for selection purposes only. When initiating the Job Processor, you can select the jobs to run based on associated frequency code. For instance, you can request to run only those jobs with a frequency code of *M*.

- ◆ **Application Code/Job Category** — These are user-defined codes that can be used to classify jobs into meaningful categories. For instance, one code can represent departmental expense reports; another code can represent vendor lists, etc. You can select the jobs to run based on associated application code/job category code or a combination of associated frequency code and application code/job category code. For instance, monthly departmental expense reports, yearly vendor lists, etc.
- ◆ **Classes** — This is another way to group jobs into meaningful categories. Jobs can be assigned to one or more classes (A-J). At run-time, you select the jobs to process by specifying the appropriate class(es).

- ◆ **Return Codes** — These are codes that are used to make the running of one job dependent on the successful, or unsuccessful, completion of another. When a job is initialized, you can indicate a code (generated by a preceding job) that the system must find before it can proceed with the current job. Likewise, in the preceding job, you can specify a code to set when the job completes successfully. You can define your own return codes (user-defined return codes) or you can use generic return codes (system options).
- ◆ **Interactive Prompt** — Allows you to display an interactive prompt window at the start and/or end of the job. This window can display a special message to the operator as well as other information pertaining to the job. The operator is given the option of continuing, skipping the next job in the stream or canceling any further processing of the job stream.
- ◆ **Selection Criteria** — Allows you to enter selection criteria to use on a permanent basis, as if the process were initiated online (only applies to Job Processor-Prepared Computron application processes).
- ◆ **Date Codes** — These codes apply only for Job Processor-Prepared Computron application processes. The date codes override the specific dates entered as part of the record selection criteria for an application. These codes eliminate the need to change record selection criteria with the passage of time. For instance, a code can indicate the first date of the month to the last day of the month or a code can designate a particular fiscal calendar.
 - ☞ *Note:* The actual date associated with generic global date codes (system options) is determined at run time. The actual date associated with user-defined global date codes is defined via Global Date Code Maintenance.

Job Processor

This function controls the running of all jobs in job processing mode. It allows an operator (without direct knowledge of the application) to accomplish the following:

- ◆ Process an entire stream of jobs, an individual job or a stream of jobs beginning with a selected job (for restarts).
- ◆ Select the jobs to process based on frequency code, application code, job category or class.
- ◆ Upon conclusion of the job(s), automatically produce a Job Process report. This is a list of the jobs successfully run, the jobs canceled or skipped by the operator, the distribution codes assigned to each job and the start and end time of each job. In addition, for all jobs that are Job Processor-Prepared Computron report functions, a record is stored in the Audit file for future use by the Audit Processor.

User Distribution List

The User Distribution List is maintained by the User Distribution List Maintenance function. The User Distribution List is a file that cross-references the IDs of users who are to receive reports with pre-defined distribution categories known as “distribution codes”. Based on the information in the User Distribution List, the Audit Processor automatically produces multiple copies of reports and distributes them to the appropriate user print file libraries. For instance, the manager of the Budget department can be assigned a

distribution code for managers and another code for the Budget department. The manager receives copies of all reports designated for managers, as well as all reports for the Budget department.

Audit Processor

This function uses the information stored in the Audit file to create multiple copies of reports (one print file is generated for each report and for each user on the distribution list for the report. The print files are distributed to the appropriate user print libraries. Thus, each user print library contains multiple files, one for each report that the user is targeted to receive.).

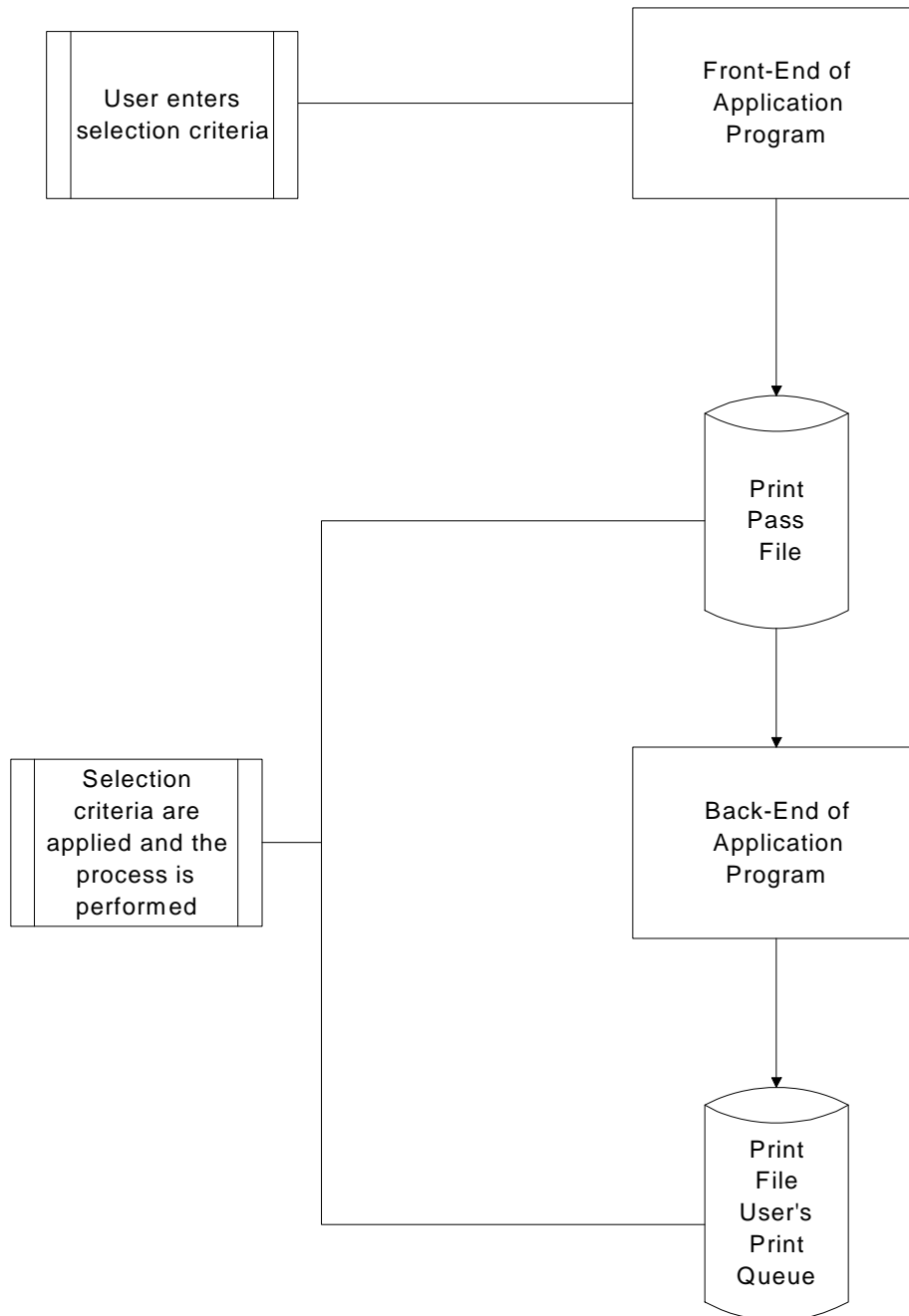
In addition this function produces the Process Audit report. This displays the following information for each user: the user's print library, the reports the user is to receive, the distribution code assigned to each report, the print files within which the reports are stored and the start and end time of report generation.

Other Programs

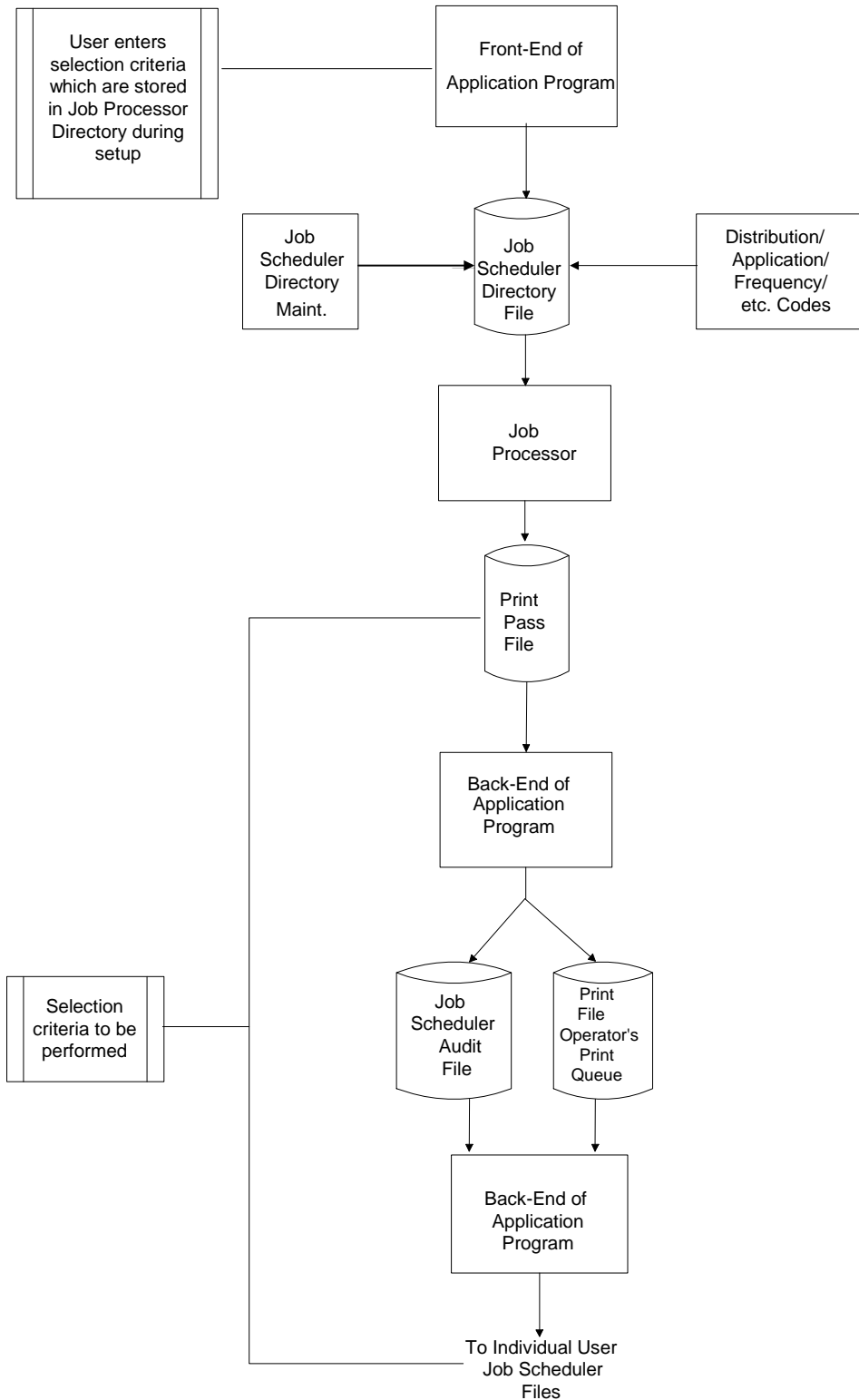
In addition to the major components described above, the Job Processing module includes many other functions. Several functions are used to create and maintain the various user-defined codes: distribution, frequency, application, job category and date codes. Inquiry/list functions are also available, allowing the user to quickly access information concerning the jobs that have been set up in the Job Scheduler Directory.

While the illustration in Fig. 3.1-1 provides a conceptual overview, it is important to understand the difference between how Job Processor-Prepared and Non-Job Processor-Prepared Computron application processes are run under Job Processing. Figs. 3.1-2 and 3.1-3 illustrate the process under each condition.

Fig 3.1-2
Computron Program — Standard Processing Flow



**Fig 3.1-3
Job Processor-Prepared Program — Job Processing Flow**



Initial Setup of the Job Processor

This subsection describes the steps required to begin working with the Job Processor.

Table 3.1-1 Steps to Establish the Job Processor Environment		
Step No.	Description	Functions
1. (Optional)	Establish Audit Processor associated codes.	User Code Maintenance Distribution Code Maintenance
2.	Establish job selection and date codes.	Global Alpha Code Maintenance Global Date Code Maintenance Global Fiscal Code Maintenance Application Code Maintenance Frequency Code Maintenance Job Category Code Maintenance
3. (Optional)	Cross-reference users with appropriate distribution codes.	User Distribution List Maintenance
4.	Enter jobs into the Job Scheduler Directory.	Job Scheduler Directory Maintenance (for Computron and non-Computron application processes) <i>Or</i> Modify Computron menus and enter via menu selection (for Job Processor-Prepared Computron application processes).

The following sections of this manual describe these steps in detail.

3.2 Job Scheduler Code List Maintenance Functions

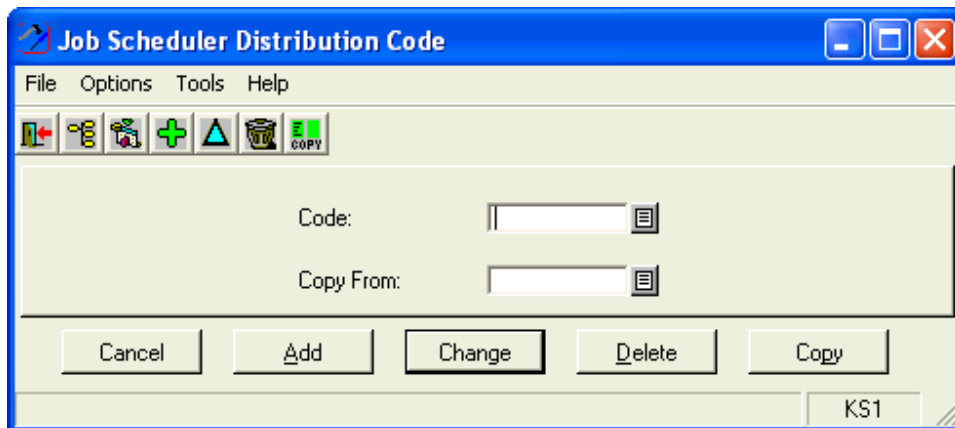
Introduction to Job Scheduler Code List Maintenance

This section explains how to maintain the Job Scheduler user-defined codes:

- ◆ distribution code
- ◆ user code (See Section 3.5, “User Code Maintenance,” for additional information.)
- ◆ global alpha code
- ◆ global date code
- ◆ global fiscal code
- ◆ application code
- ◆ frequency code
- ◆ job category code.

Code Maintenance Window (KS1)

From the Code Maintenance Selection Window (KS1) enter the appropriate code to maintain (Q-^{Mark}) and click the desired maintenance option (Add, Change, Delete, or Copy). The Code Maintenance Window (DS1) displays.



Code Maintenance Window (KS1)

Code Maintenance Window (DS1)

The DS1 window is used to enter the details about each code. Although the fields that display for each Code Maintenance Window (DS1) vary depending on the code under maintenance, the following fields are common to all.

Code List

Enter the name and description of the code list to maintain. (This will vary depending on the maintenance function you select.)

Code

Enter the user assigned code to maintain. This field appears on Q-^{Mark} windows.

Active

Enter the status of the code to maintain. This field appears on Q-^{Mark} windows. This is a required field. Valid entries are:

- The code is active and can be used.
- The code is inactive.

Default:

Record Class

Enter the record class to assign to this code. This field is used in combination with parameters established in the User Profile Security file to restrict this record to a specified class of users. Valid entries are A–Z.

Security Level

This security field is used in combination with data established in the User file to restrict the use of this code. If a value is entered for this field, this record is available only to users with an equal or higher security level. Valid entries are 0–9.

Owner Code

This field is contingent upon parameters established in the User Profile Security file. Only users with this ownership code can access this record.

Descr. (Name)

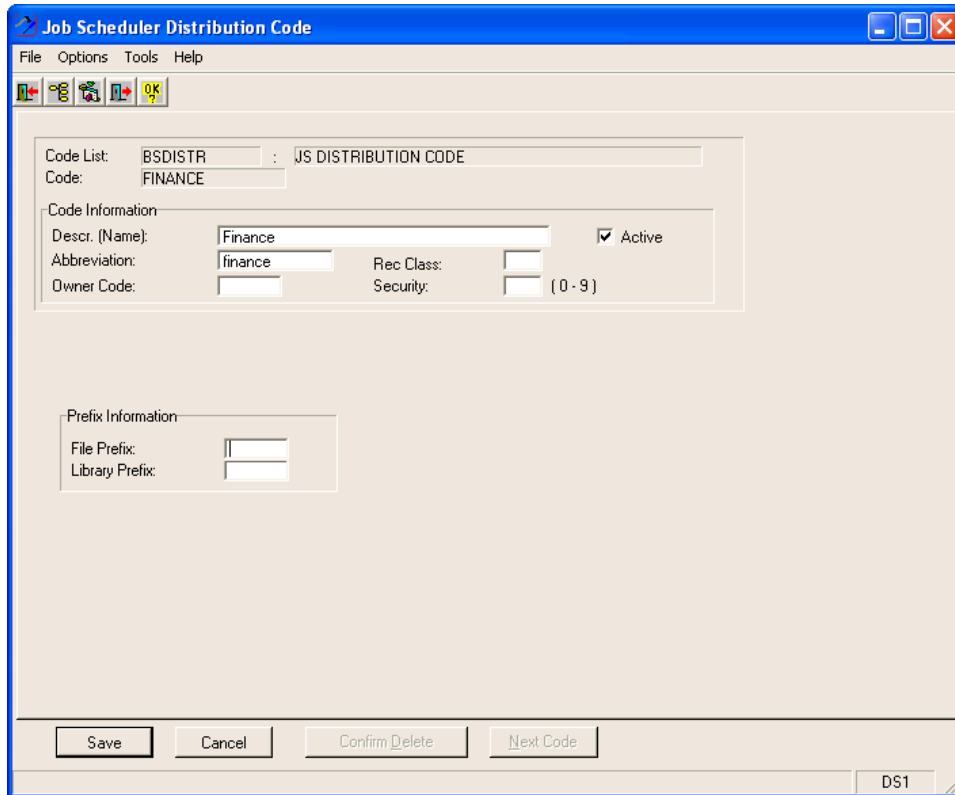
Enter a name up to thirty alphanumeric characters that describes the appropriate code. This field appears on Q-^{Mark} windows.

Abbreviation

Enter an abbreviation of up to ten alphanumeric characters for the code. This abbreviation is used on reports when the description is too long to print in its entirety. This field appears on Q-^{Mark} windows.

Distribution Code Maintenance

Distribution codes are used to establish logical categories or groupings for reports. For instance, distribution codes can be established for the accounts payable department, accounts receivable department and senior managers.



Distribution Code Maintenance Window (DS1)

File Prefix

Not currently implemented. Leave blank.

Library Prefix

Not currently implemented. Leave blank.

User Code Maintenance

See Section 3.5, “User Code Maintenance,” for additional information.

Global Alpha Code Maintenance

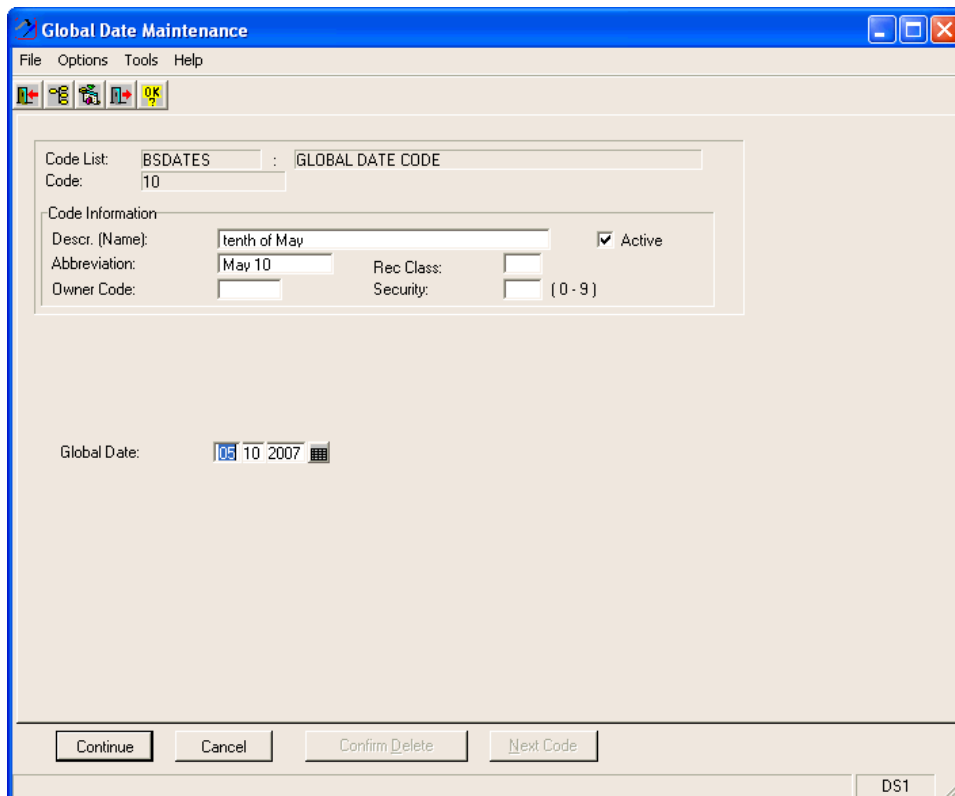
This code is not in use.

Global Date Code Maintenance

This code is used to control the selection of jobs for processing.

User-defined global date codes can be used only in conjunction with Job Processor-Prepared Computron application processes. Each user-defined global date code represents a specific date. When adding a job to the Job Scheduler Directory, you can enter a user-defined global date code in one of the appropriate date code fields on the Job Scheduler Directory Maintenance Window (BS1). The code is used in lieu of the date specified as part of the usual record parameters for this application process.

The actual date associated with the global date code can be changed at any time. For instance, you can establish a code CPE to represent the current period ending date — 12/31/2006. Then you only have to modify the actual date (12/31/2006) via the Global Date Code Maintenance function each time you wish to change the current period. If one global date code is used by several different jobs, changing the global date code once via the Global Date Code Maintenance function, affects the dates used for all the jobs.



Global Date Code Maintenance Window (DS1)

Global Date

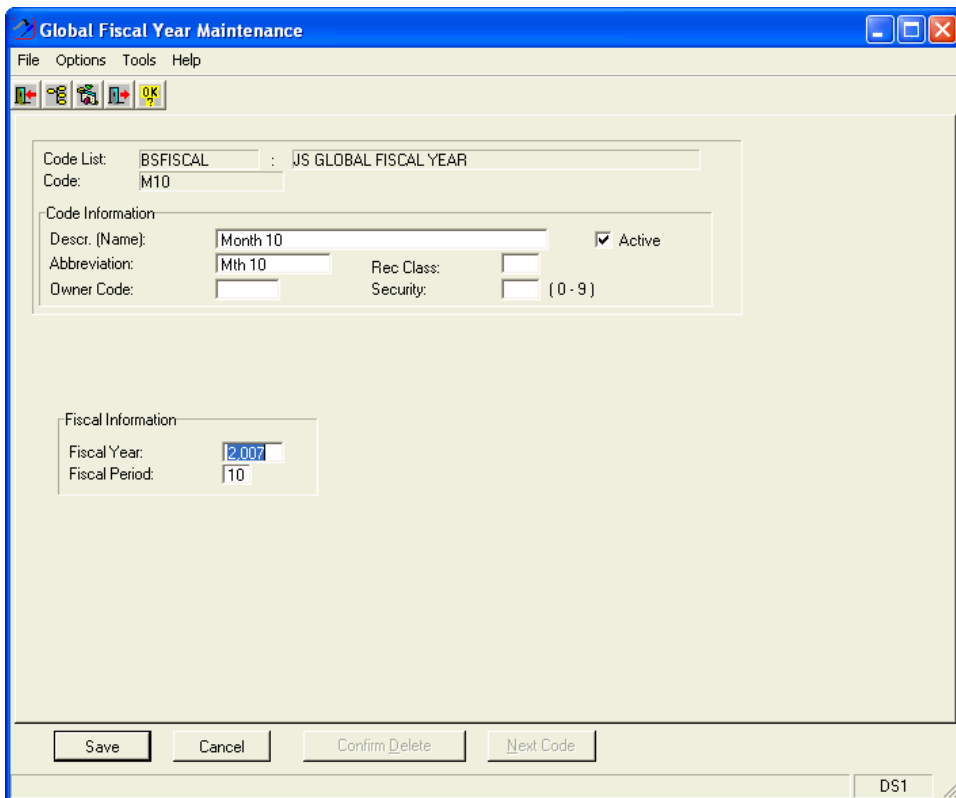
Enter the date to associate with this code. You can use the calendar icon next to the date field to select a date.

Global Fiscal Code Maintenance

Global fiscal codes are used in the same manner as global date codes (see above). The only difference is that the global fiscal code is associated with a fiscal year and period, rather than a specific day of the year.

Notes:

- ◆ How the system interprets the global fiscal code depends on the calendar in use by the Computron application system.
- ◆ The General Ledger application accommodates multiple calendars and period-ending dates are user-defined.
- ◆ Generic global fiscal codes (system options) are not available.



Global Fiscal Code Maintenance Window (DS1)

Fiscal Year

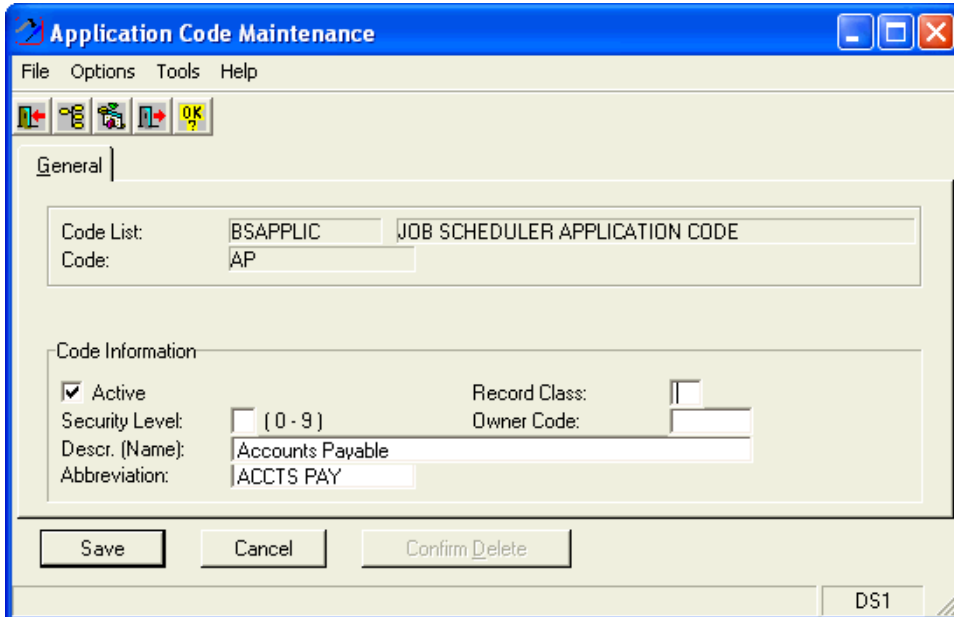
Enter the fiscal year to associate with this code.

Fiscal Period

Enter the fiscal period to associate with this code.

Application Code Maintenance

Application codes provide a means of classifying jobs. For instance, one code can represent departmental expense reports; another code can be for vendor lists, etc. You can select the jobs to run based on the application code.

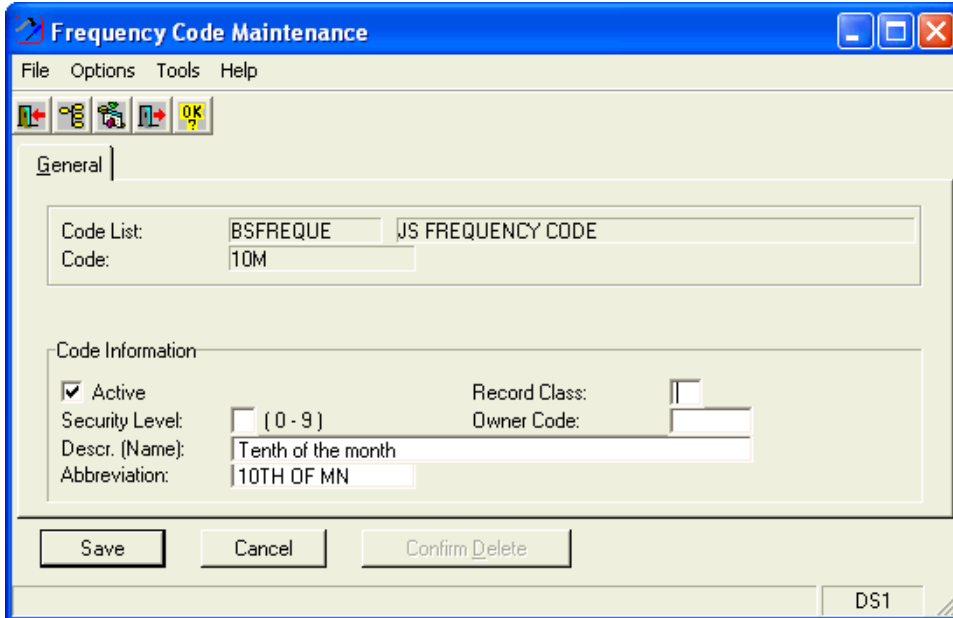


Application Code Maintenance Window (DS1)

Frequency Code Maintenance

The Job Processor uses user-defined frequency codes as additional selection criteria. Generic frequency codes are used specifically to indicate how often a job should run (i.e., the frequency).

When a job is added to the Job Scheduler Directory, it is assigned a frequency code. This code can be generic (i.e., one of the system options) or user-defined. (e.g., D for daily, M for monthly etc.). When initiating the Job Processor, you can select the jobs to run based on the frequency code. For instance, you can elect to run only those jobs with a frequency code of M.

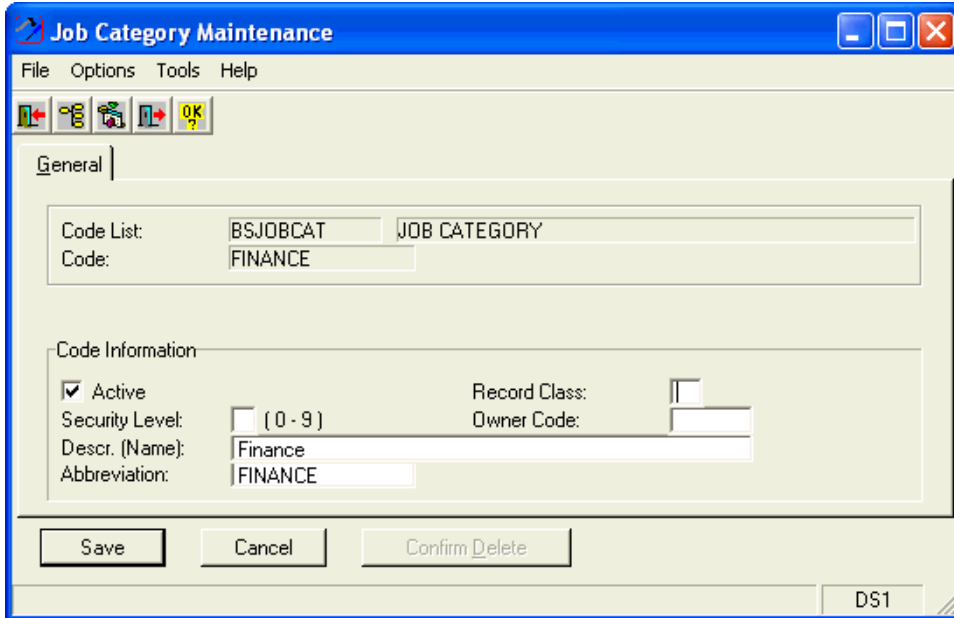


The screenshot shows a window titled "Frequency Code Maintenance" with a menu bar (File, Options, Tools, Help) and a toolbar. The "General" tab is selected. The "Code List" field contains "BSFREQUE" and "JS FREQUENCY CODE". The "Code" field contains "10M". The "Code Information" section includes a checked "Active" checkbox, a "Security Level" dropdown set to "(0 - 9)", a "Record Class" dropdown set to "1", and an "Owner Code" field. The "Descr. (Name)" field contains "Tenth of the month" and the "Abbreviation" field contains "10TH OF MN". At the bottom, there are "Save", "Cancel", and "Confirm Delete" buttons, and a "DS1" label in the bottom right corner.

Frequency Code Maintenance Window (DS1)

Job Category Code Maintenance

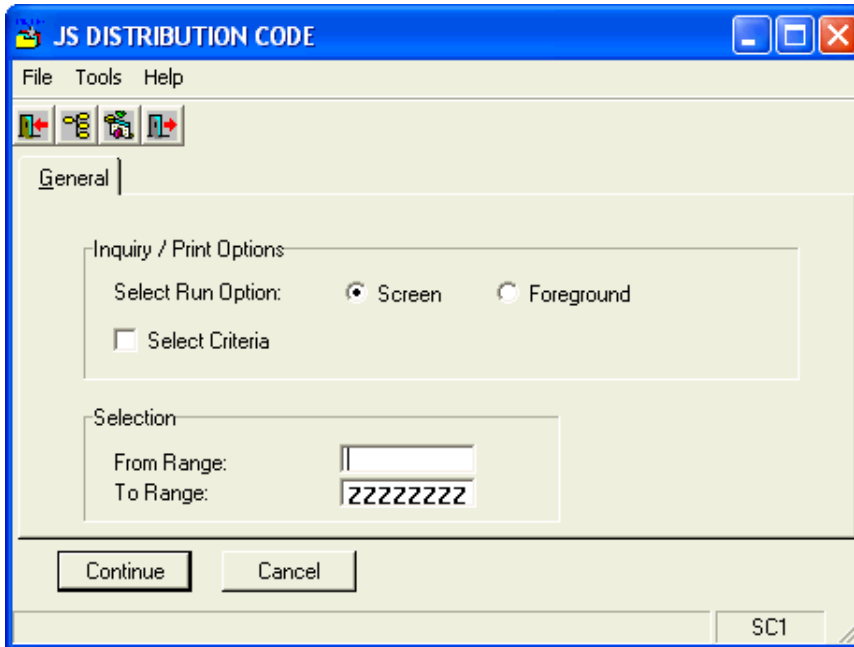
Job category codes are similar to application codes. These codes provide an additional means of classifying jobs.



Job Category Code Maintenance Window (DS1)

Inquiry / List Functions

For every code maintenance function available, a corresponding list function is also available. Below is a sample of a typical List Selection Window (SC1). Through this window, you can select the data to include in the list. The data that displays may differ for each function, but emulates the data entered on the key window (KS1) for the related maintenance function.



List Selection Window (SC1)

Select Run Option

This is the processing option that either produces the online inquiry or the printed report. Valid options are:

Screen – This option enables you to review a list of the data online, as well as to display detail for each item on the list.

Foreground – This option produces a print file, processing the request in foreground. Use of this option restricts the use of this window for other processing.

Default: Foreground.

Select Criteria

This field determines whether to use a query window (DS99) to specify additional selection criteria. Valid options are:

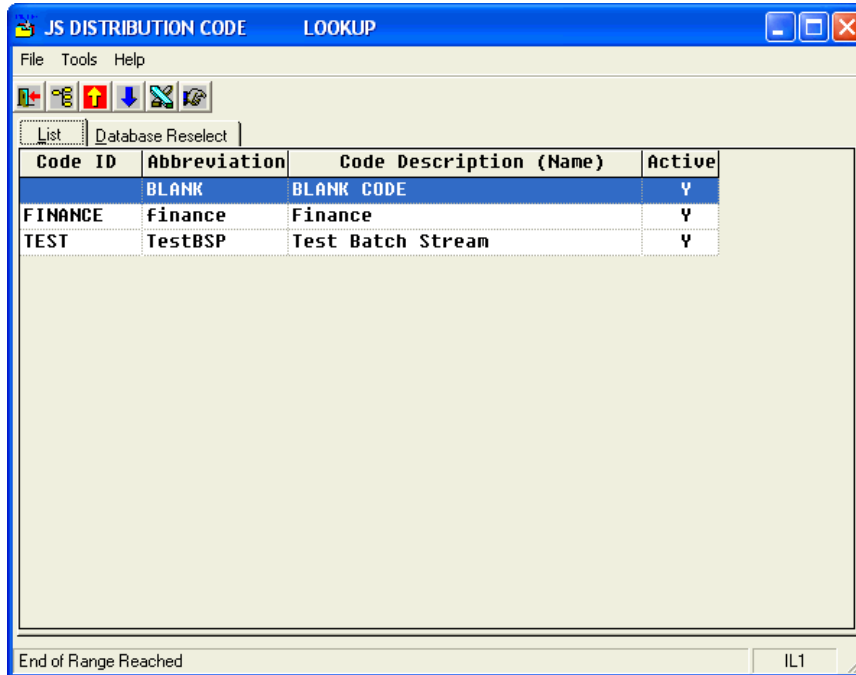
- Displays the Select Query Window (SC99), which allows you to further define the selection criteria.
- Do not allow modification of the selection criteria.

Default:

The remaining fields differ, depending on the List function selected.


Screen Option

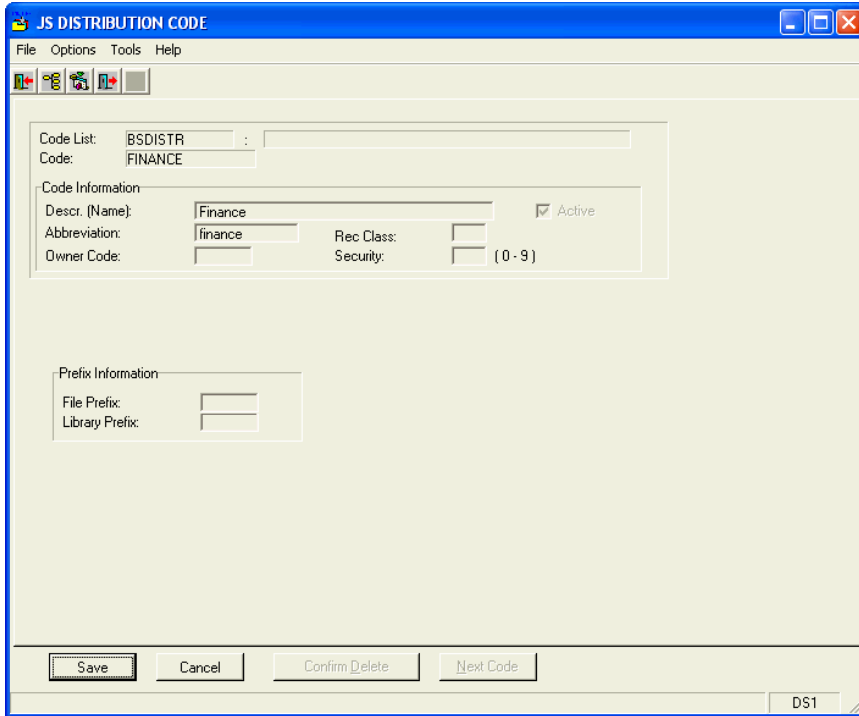
When the Screen option is selected, an inquiry window displays.



Inquiry Window Display (IL1)

To display the details of a listed code, you can either:

- ◆ highlight it and click the Choose toolbar button 
- ◆ double-click on the desired code.



The screenshot shows a window titled "JS DISTRIBUTION CODE" with a menu bar (File, Options, Tools, Help) and a toolbar. The main area contains the following fields:

- Code List: BSDISTR
- Code: FINANCE
- Code Information:
 - Descr. (Name): Finance Active
 - Abbreviation: finance
 - Owner Code: []
 - Rec Class: []
 - Security: [] (0-9)
- Prefix Information:
 - File Prefix: []
 - Library Prefix: []

At the bottom, there are buttons for Save, Cancel, Confirm Delete, and Next Code. The status bar at the bottom right shows "DS1".

Inquiry Window Detail Display (DS1)

Print Options

The following sample report is available for a Job Processor inquiry/list function:

Sample Job Processor Code Listing

```

*   Computron Software, LLC   ***** Thu May 03, 2007 11:20 AM *
*
*   --- Job Scheduler Distribution Code ---
*
*   ***** Page 1 *

```

Code ID	Abbreviation	Code Description (Name)	Active
	BLANK	BLANK CODE	Y
FINANCE	finance	Finance	Y
TEST	TestBSP	Test Batch Stream	Y

3.3 Job Scheduler Directory Maintenance

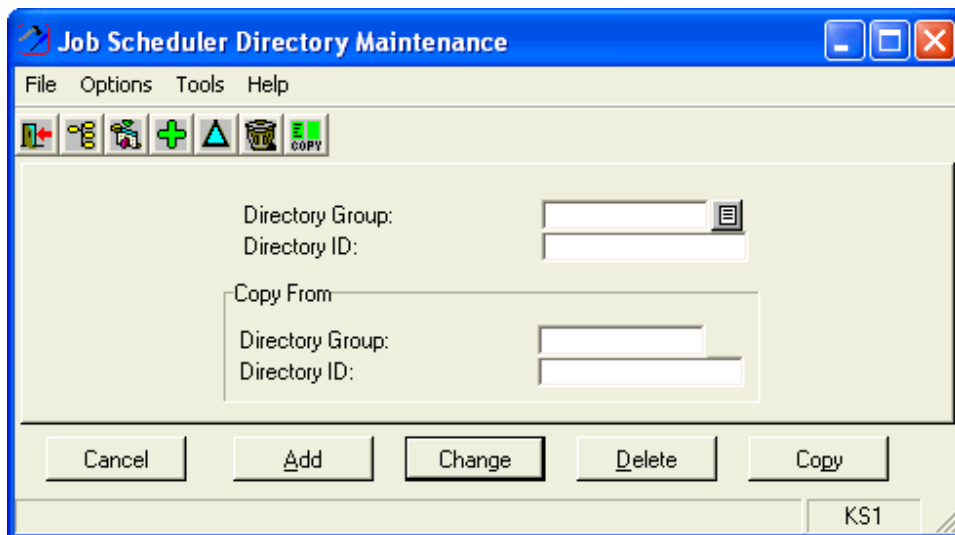
Introduction to Job Scheduler Directory Maintenance

Use this function to maintain the Job Scheduler Directory. This file contains the information used by the Job Processor to control the running of jobs in the process stream.

Note: If the job is a Job Processor-Prepared Computron function, Job Scheduler Directory Maintenance is not used to add the job to the Job Scheduler Directory. Instead, you should run the Computron program from the appropriate application system menu. If the menu has been set up properly, additional maintenance windows should appear, allowing you to add the job to the Job Scheduler Directory. Once the job has been added to the directory, Job Scheduler Directory Maintenance can be used to maintain the information that has been recorded.

This function is used to create Non-Computron Program/Non-Job Processor-Prepared Computron Program items in the list of jobs to execute.

Job Scheduler Directory Maintenance Window (KS1)



Job Scheduler Directory Maintenance Window (KS1)

Directory Group (Q^{Mark})

Each job must be assigned to a group within the directory. Enter a directory group identifier up to six alphanumeric characters.

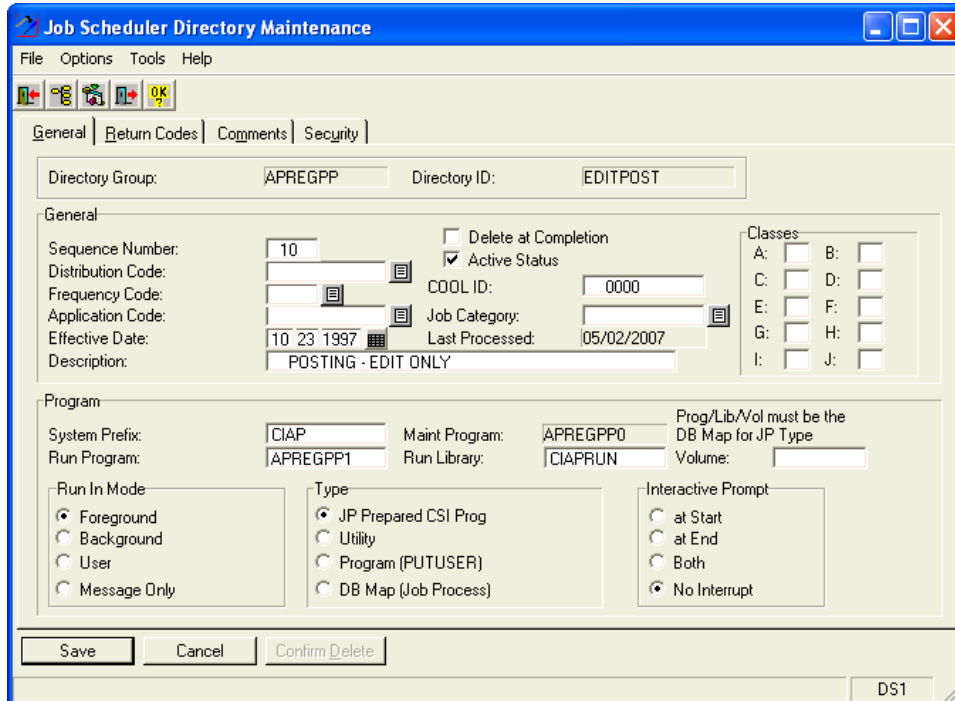
Note: When adding a job to the directory, do not use one of the directory groups associated with a Job Processor-Prepared Computron program.

Directory ID (Q-^{Mark})

Within each directory group, each job has a unique identifier. Enter the identifier for the job up to ten alphanumeric characters.

From the Job Scheduler Directory Maintenance Window (KS1), enter the appropriate directory group (Q-^{Mark}), and directory ID (Q-^{Mark}), and click the desired maintenance option (Add, Change, Delete, or Copy). The Job Scheduler Directory Maintenance Window (DS1) displays

Job Scheduler Directory Maintenance Window (DS1)



The screenshot shows the 'Job Scheduler Directory Maintenance' window with the following fields and options:

- Directory Group: APREGPP
- Directory ID: EDITPOST
- General section:
 - Sequence Number: 10
 - Distribution Code: [empty]
 - Frequency Code: [empty]
 - Application Code: [empty]
 - Effective Date: 10 23 1997
 - Description: POSTING - EDIT ONLY
 - Delete at Completion:
 - Active Status:
 - COOL ID: 0000
 - Job Category: [empty]
 - Last Processed: 05/02/2007
 - Classes: A, B, C, D, E, F, G, H, I, J (all unchecked)
- Program section:
 - System Prefix: CIAP
 - Run Program: APREGPP1
 - Maint Program: APREGPP0
 - Run Library: CIAPRUN
 - Volume: [empty]
 - Run In Mode: Foreground, Background, User, Message Only
 - Type: JP Prepared CSI Prog, Utility, Program (PUTUSER), DB Map (Job Process)
 - Interactive Prompt: at Start, at End, Both, No Interrupt

Buttons at the bottom: Save, Cancel, Confirm Delete. The window title bar shows 'DS1'.

Job Scheduler Directory Maintenance Window (DS1)

Sequence Number

This number determines the order in which the jobs are run. The job with the lowest number is run first. If two or more jobs have the same number, the directory group and the directory ID determine the order.

Delete at Completion

This field applies to jobs that only run once. Valid options are:

- Delete the job from the Job Scheduler Directory when it has successfully completed.
- Retain the job in the Job Scheduler Directory.

Default:

Classes

A job can be assigned to one or more categories known as classes. Enter any value 0-9 and A-Z within each class (A-J). At run time, select the job to process by specifying the class and associated value.

Example: A job can be assigned to classes A, B and J for processing as follows:

A	B	C	D	E	F	G	H	I	J
Z	1	*	*	*	*	*	*	*	Q

At run time, you can elect to run only those jobs with a Z in Class A. However, more typically, the same value is entered in each Class field, and always selects on that character.

Example:

A	B	C	D	E	F	G	H	I	J
X	X	*	*	*	*	*	*	*	X

Enter X in the Class A field of the Job Processor to process all jobs assigned to class A.

Distribution Code (Q-Mark)

This code determines which users receive copies of reports.

Validation: This must be a valid code previously established via Distribution Code Maintenance.

Active Status

This field determines whether the Job Processor runs the job. Valid options are:

- Process the job.
- Do not process the job.

Default:

Frequency Code (Q-Mark)

This code is used to schedule the running of the job. This field is used in conjunction with either standard system options, refer to Table 3.3-1 or with user-defined codes accessed via Q-Mark. The system options allow you to indicate that the job is to be run on a certain day of the week (Monday, Tuesday, etc.), a certain day of the month (1st, 2nd, etc.) or on the nth Monday (Tuesday, Wednesday, etc.) of the month. The Job Processor compares the frequency code to the processing date entered at runtime to determine whether the job should run. For a list of available system options, refer to Table 3.3-1.

You can also specify a user-defined frequency code established via Frequency Code Maintenance, such as D for daily, M for monthly, and so on. These codes are used for selection purposes only. When initiating the Job Processor, you can select the jobs to run based on frequency code. For instance, request only those jobs with a code of M.

Note: if this field is left blank, the Job Processor always selects the job.

Table 3.3-1 Frequency Codes — System Options	
Code	Description
#W n	n = day of week (#W1 = Sunday, #WN = Sunday)
# nn	nn = day of month (#32 = last day of month)
* wn	w = week of month n = day of week (*67 = Last Saturday of month)

Examples:

Code	Description
#W1/#WN	Sunday
#W2/#WM	Monday
#W3/#WT	Tuesday
#W4/#WW	Wednesday
#W5/#WH	Thursday
#W6/#WF	Friday
#W7/#WS	Saturday
#01	First day of month
#02	Second day of month
.	
.	
#32	Last day of month
*11	First Monday of month
*12	First Tuesday of month
*27	Second Sunday of month
*67	Last Saturday of month

☞ *Note:* The Job Processor runs only those jobs whose frequency codes match the processing date. If the processing date is Wednesday, 05/30/2007, jobs with the

following frequency codes are allowed to run: #W4 (Wednesday), #30 (Wednesday 30th day of month), *54 (5th Wednesday of month).

COOL ID

This is the ID of the report that will be produced. This ID is used when the report is moved from the Computron print queue to the AXS-One Central archive.

Application Code (Q-^{Mark})

This is a user-defined code used to classify jobs. When initiating the Job Processor, select the job to process by specifying the application code.

Job Category (Q-^{Mark})

This is a user-defined code used to classify jobs. When initiating the Job Processor, select the job to process by specifying the job category code.

Effective Date

This is the earliest date the job can run.

Validation: The Job Processor compares the effective date to the processing date entered at run time to determine whether or not the job should run. As a default, the system displays the current date.

☞ *Note:* Enter the earliest processing date for which the job is likely to run in the Effective Date field. This is because although the function is added to the Job Scheduler Directory as of the current date, you may wish to run the function as of a date in the previous month. The effective date should therefore be set to the earlier date.

Last Processed

This field displays the date the job was last run. It is updated by the system and cannot be modified.

Description

This is a description of the job, up to thirty alphanumeric characters. It appears on Q-^{Mark} windows and on the Job Scheduler Directory report.

System Prefix

If the program to run is part of a Computron application, this field is used to identify the standard library prefix for that application. For example, CIGL is a prefix used for all libraries in the Computron General Ledger system. This field is for reference purposes only.

Maint Program

This field is updated by the system and cannot be modified. It applies only to Job Processor-Prepared Computron programs. These programs have been divided into two parts: the front-end (used for record selection criteria entry — maintaining the Job Scheduler Directory) and back-end (runs program). The name of the front-end portion displays in this field.

Run Program / Run Library / Volume

These fields indicate the location, file/library/volume, of the program to run. If the volume is left blank, the system uses the volume associated with the Job Processor.

- ☞ *Note:* If the purpose of the job is to direct the Job Processor to switch to a different database to process another stream of jobs (the Type field is set to D, see Type field), these fields must show the file/library/volume of the new database map.

Run in Mode

This is the processing option that determines how to process the job. Valid options are:

Foreground – This option processes the job in foreground. Use of this option restricts the use of this window for other processing.

Background – The job is placed on the queue for subsequent processing. The terminal is available for other functions while the job is running.

Default: Foreground.

- ☞ *Note:* The Foreground and Background options override whatever run mode is selected by the operator when initiating the Job Processor.

User – This option allows the operator to select the run mode when initiating the Job Processor. The operator can select from Foreground, Background or Depends on Job. If the operator selects the Depends on Job option when initiating the Job Processor, the run mode is Foreground.

Message Only – Select this option if the purpose of the job is to display a message to the operator. See Interactive Prompt field.

- ⚠ *Warning:* Care should be taken when selecting the option for the Run in Mode field as return code dependencies are affected. The following restrictions apply:

Foreground – If an individual job within the job stream is submitted for processing in foreground, the next job in the job stream is not submitted until the former job is processed.

Background – If an individual job within the job stream is submitted for processing in background, the next job can be submitted for processing in either foreground or background immediately. This can disable the latter job from validating any return code results from the first job.

User/Background – If the entire job stream is submitted for processing to background, each job is run in background and submitted only on completion of the prior job.

Type

This field indicates the job type. Valid options are:

JP Prepared CSI Prog (O) – The job is a Job Processor-Prepared Computron program. The system automatically enters O when the job is added to the Job Scheduler Directory. This is a non-modifiable field.

- ☞ *Note:* The job also passes parameters from the front-end program via the Print Pass file.

Utility (U) – The job is a non-Computron utility or program. This is the default for all jobs added via Job Scheduler Directory Maintenance.

Program (PUTUSER) (P) – The job is a Computron program to run interactively.

- ☞ *Note:* The job issues a “PUTUSER” PUTPARM.

DB Map (Job Process) (D) – This job instructs the Job Processor to switch to a different database and begin processing another stream of jobs. The system processes the jobs in the Job Scheduler Directory for the other database.

☞ *Note:* If this option is selected, the Run Program/Library/Volume fields (see above) must contain the next database to process.

Interactive Prompt

This option displays a special Interactive Prompt window at the start and/or end of the job. This window can include a message to the operator in addition to general information about the job and prompts that allow the operator to continue or cancel processing. Valid options are:

at Start – Display the interactive prompt at the start of the job.

at End – Display the interactive prompt at the end of the job.

Both – Display an interactive prompt at both the start and end of the job.

No Interrupt – Do not display an interactive prompt.

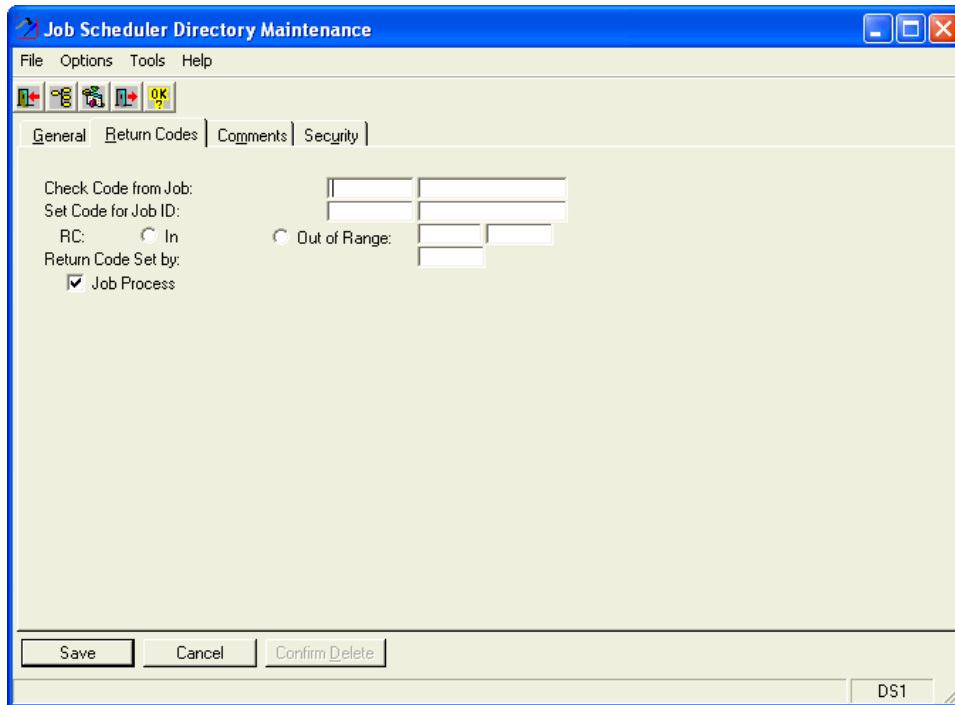
Default: No Interrupt.

☞ *Notes:*

- ◆ Interactive prompts can be suppressed at run time. For details, refer to Section 3.8, “Job Processor.”
- ◆ If at Start, at End, or Both is selected, an additional window (DS2) displays to provide for message entry.
- ◆ If the job is a Job Processor-Prepared Computron program, you cannot display a message as part of the interactive prompt. However, you can establish a “Message Only” job to run either before or after the job in question.

Job Scheduler Directory Maintenance Window (DS1) Return Codes tab

The DS1 Return Codes tab displays when you click the Return Codes tab:



Job Scheduler Directory Maintenance Window (DS1) - Return Codes

Note: Return codes make the running of one job dependent upon the successful completion of another.

Check Code from Job

These fields apply if the current job depends on the successful completion of a preceding job. Use these fields to identify the preceding job. For instance, enter the directory group and directory ID (or whatever code was used by the preceding job in the Set Code for Job ID field).

Set Code for Job ID

Use these fields to identify the current job. The succeeding job reads this return code to determine if processing should proceed. The key to the record in the Return Code file is the job ID established here. These fields apply if a succeeding job depends on the successful completion of the current job. For instance, enter the directory group and directory ID. When the current job completes, the return code is written to the Return Code file to be read by a subsequent job.

RC

This prompt is made up of two questions. Set the In or Out of Range radio button and enter a range of return code values (From — To). The system must find a return code within/outside the range stipulated in this field before it can proceed with the current job. This field allows

you to indicate a range of values for the return code and whether the return code must be within or outside of this range. Valid values for the radio buttons are:

In – The return code must be within this range.

Out of Range – The return code must be outside this range.

Default: In

Return Code Set by

This is the return code to set if the current job is processed successfully (i.e., is not canceled by the operator). This return code is stored in the Return Code file along with the job ID established in the Set Code for Job ID field. You can enter any alphanumeric code, including blanks or a special keyword — LRC# (Link Return Code).

☞ *Note:* If all numeric values are entered in this field, they are right justified when written to the Return Code file.

Example:

RC Set To	Stored As
15	0015
150	0150
1500	1500

It is important to be aware of how return codes are written to the Return Code file, to enable you to establish an appropriate value that the system can test against in the RC (From — To) fields.

☞ *Note:* Enter LRC# (Link Return Code): to retrieve the system-defined return code set by the current program and write that value to the Return Code file. The dependent job should be configured to interpret this return code value and proceed accordingly. Table 3.3-2 lists each of the system-defined return codes.

Return Code	Description
888	Process canceled by operator.
889	Process canceled by CSDISASTER.
997	Process canceled due to submaroon failure.
999	Link failed.
2101	Previous job was not run. Job Processor record not found. Job skipped, (Job Processor-prepared Computron programs only.)
8101	Previous job in the job stream was canceled.

Table 3.3-2 Return Codes — System-Defined	
Return Code	Description
1101	Return code not in range checked. Job skipped. (Job Processor-prepared Computron programs only.)
0/103	OK.
5/105	OK, but no records found.
102	OK, but warnings found.
101	Errors encountered.
111	Errors encountered evaluating the selection criteria or other values.
123	Print Pass record not found.
124	Operator has no access to program or report.
125	Program or report not available. (Locked by another user.)
126	Program or report not found.
131	Program was interrupted by user.

The first digit of a return code varies from application to application: consequently, only the two right-most digits of a return code are constant. Since only the last two right-most digits remain constant, only those digits should be tested in scripts, etc.

Masking can be used to specify the return code range so that only the last two characters are validated. For instance, the return code range: From **03 To 0003, replaces the first two characters (denoted by *) in the From field with the value of the same characters in the To field. The masked characters (as specified in the From field) are similarly replaced in the return code field of the Return Code file before a compare is performed; therefore, a return code value of 0103 in the Return Code file becomes 0003.

Job Process

This field indicates whether the user-defined return code should be written to the Return Code file by Job Processing. Valid entries are:

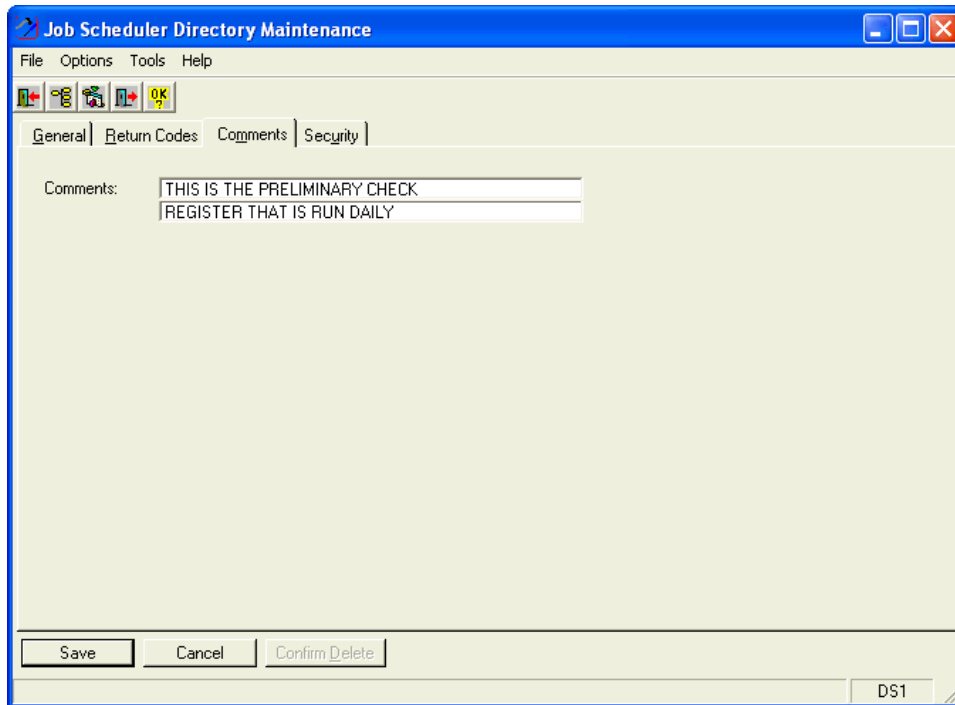
- Write the user-defined return code to the Return Code file.
- Do not write the user-defined return code to the Return Code file.

Default:

Note: When you enter LRC# in the RC Set To field, a record is always written to the Return Code file, regardless of the option selected in this field.

Job Scheduler Directory Maintenance Window (DS1) Comments tab

The DS1 Comments tab displays when you click the Comments tab:



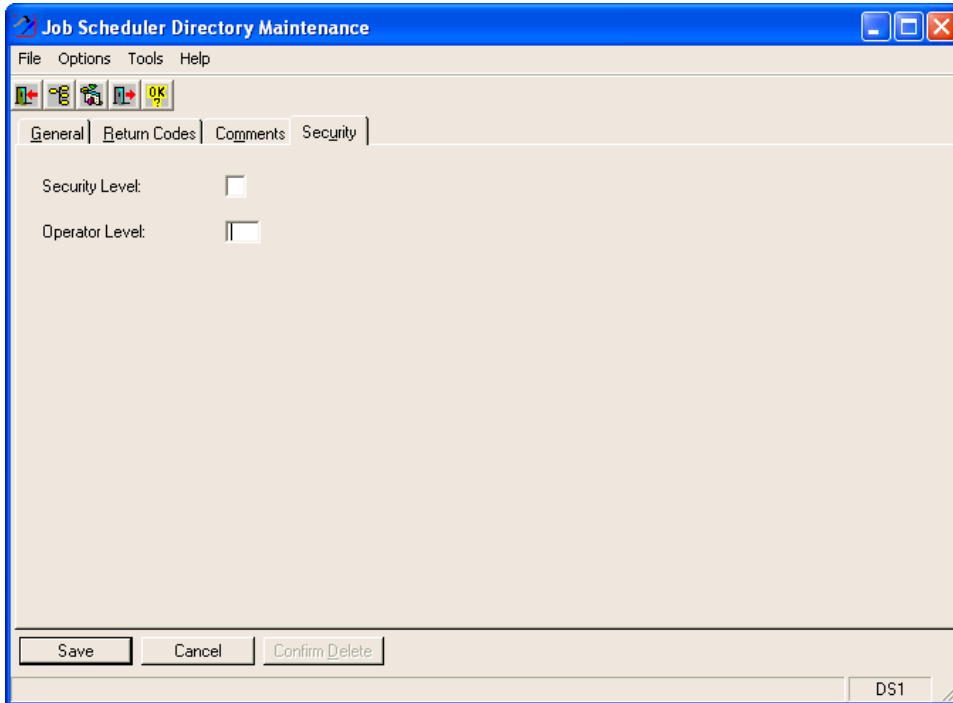
Job Scheduler Directory Maintenance Window (DS1) – Comments

Comments

Both thirty-position fields can be used to enter a comment about the job. This does not display to the operator. For the Comments to display to the operator see the Job Scheduler Directory Maintenance Window (DS2), later in this section.

Job Scheduler Directory Maintenance Window (DS1) Security tab

The DS1 Security tab displays when you click the Security tab:



Job Scheduler Directory Maintenance Window (DS1) -Security

Security Level

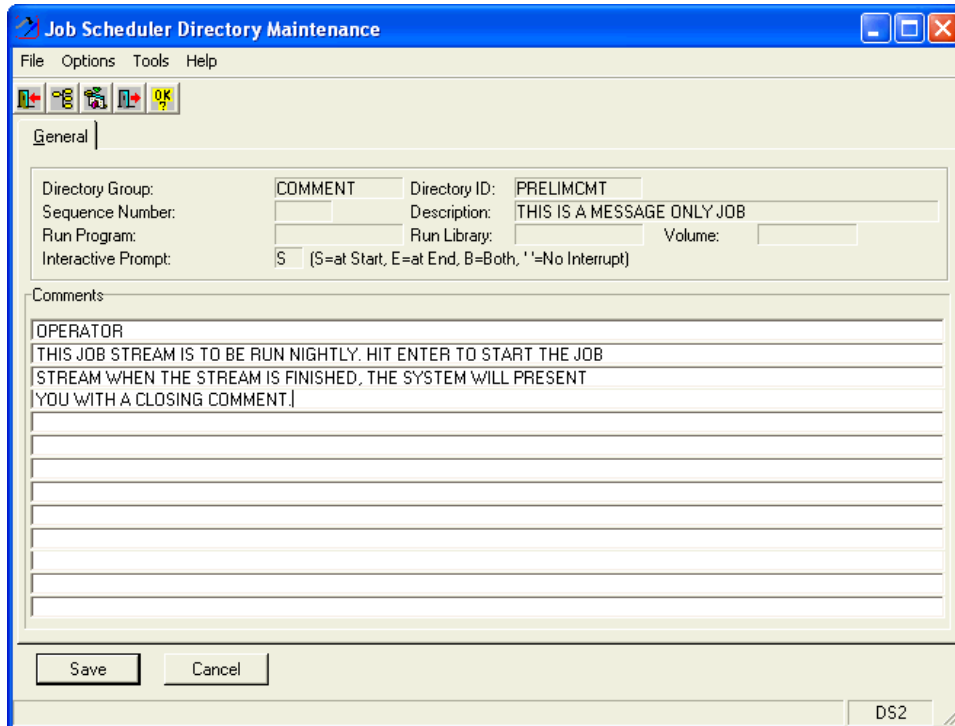
This field is not in use.

Operator Level

This field is not in use.

Job Scheduler Directory Maintenance Window (DS2)

The DS2 window displays if the job is a non-Job Processor-Prepared Computron job and the Interactive Prompt field is set to at Start, at End or Both.



Job Scheduler Directory Maintenance Window (DS2)

Comments

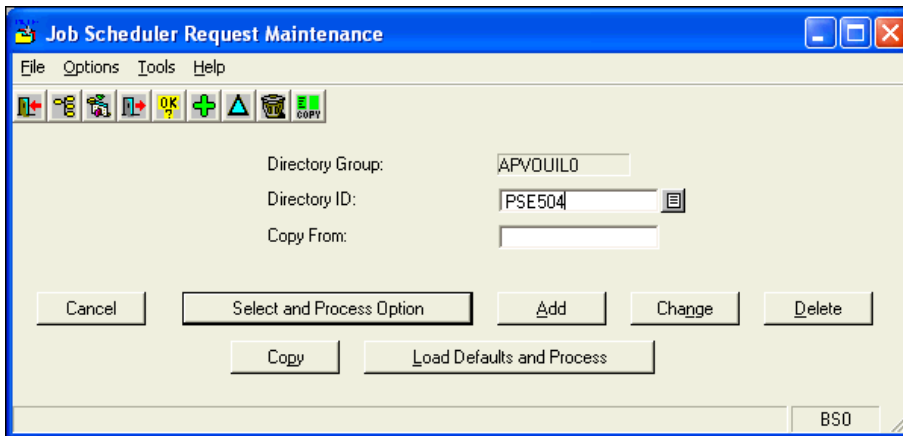
Use this area to enter the message to display on the operator's window at run-time.

Job Processor-Prepared Computron Programs

If the job is a Job Processor-Prepared Computron program, it can be added to the Job Scheduler Directory by running the program from the appropriate application system menu. Assuming the menu has been properly set up, two additional windows appear. See Section 3.11, "Menu System Integration," for additional information on menu tailoring.

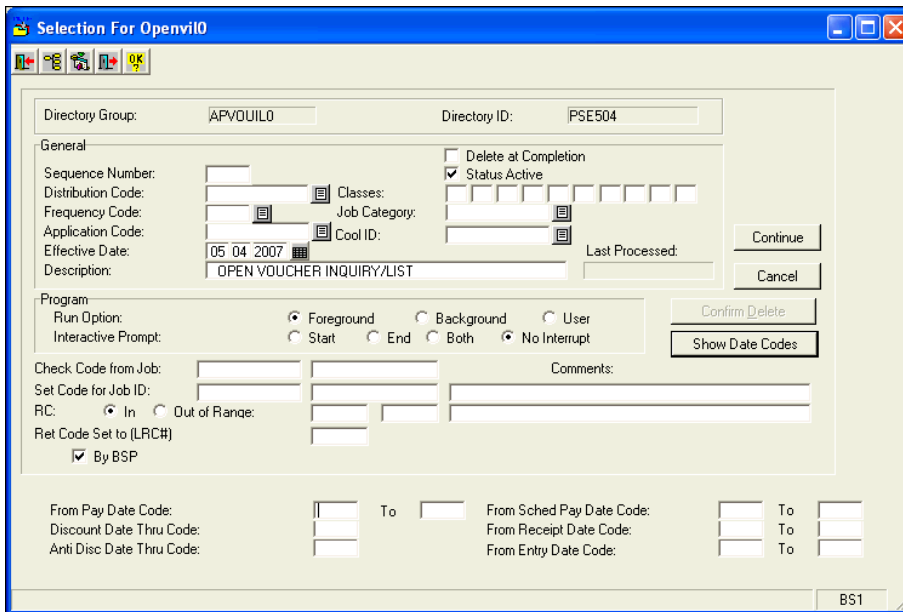
The first window — Job Scheduler Maintenance Window (BS0) prompts for a directory ID and the selection of a maintenance option (Add, Change, Copy, or Delete). The directory group is automatically assigned by the program and cannot be modified.

The second window — Job Scheduler Maintenance Window (BS1) is a modified version of the Job Scheduler Directory Maintenance Window (DS1). While most of the options on this window are the same as those described on the preceding pages, the differences between them are detailed below.



Job Scheduler Maintenance Window (BS0)

The Process options (grayed out above) are not displayed if the menu has been set up with the M option, see Section 1.4, “Menu File Maintenance,” for additional information. Use of this option effectively restricts any user from accessing the function directly (i.e., outside of the Job Processing environment).



Job Scheduler Maintenance Window (BS1)

Some fields do not display; the information is supplied by the program. These fields are:

- ◆ Security Level
- ◆ Operator Level
- ◆ System Prefix
- ◆ Maintenance program
- ◆ Run Program
- ◆ Run Library
- ◆ Run Volume
- ◆ Type

Additional fields may display on the window to allow you select date codes. The date codes entered in this window override the specific dates or periods entered as part of the record selection criteria for the program. Enter either generic date codes or user-defined date codes (set up in Global Date Code Maintenance or Global Fiscal Code Maintenance). If these fields are left blank, the program uses the specific dates entered as part of the usual record selection.

It is important to enter the appropriate type of date code. For instance, if the current job prompts you to supply a fiscal period and year, a code established via Global Fiscal Code Maintenance should be selected, not a code established via Global Date Code Maintenance. Use the Q-^{Mark} facility to access a list of user-defined codes. Tables 3.3-3 lists each of the system options and provide an explanation of each.

Table 3.3-3 Date Codes — System Options	
Code	Description
TOD	Today's date
YES	Yesterday's date
BOY	First day of current year
PYB	First day of prior year
NYB	First day of next year
BOM	First day of current month
PMB	First day of prior month
NMB	First day of next month
BOW	First day of current week (Sunday)
PWB	First day of prior week (Sunday)
NWB	First day of next week (Sunday)
TOM	Tomorrow's date
EOY	Last day of current year
PYE	Last day of prior year
NYE	Last day of next year
EOM	Last day of current month
PME	Last day of prior month
NME	Last day of next month
EOW	Last day of current week (Saturday)
PWE	Last day of prior week (Saturday)

Table 3.3-3 Date Codes — System Options	
Code	Description
NWE	Last day of next week (Saturday)
CCP	Current month/year
CPP	Prior month/year
CNP	Next month/year
CYB	First month of current year

Sample Application

Return Codes

In the following example, one of the jobs to run from the Job Processor entails the posting of a large volume of transactions to the General Ledger. However, it has been determined that before this job can run, a successful backup should be performed of all files in the General Ledger system.

Set up two jobs in the Job Scheduler Directory; one to perform the backup and one to perform posting (via one of the Transaction Posting programs in Computron's General Ledger system). Enter the following information:

Job 1 (Backup)	
Directory Group	UTILS
Directory ID	BACKUP
Set Code for Job ID	UTILS BACKUP
RC Set To	OK
By BSP	<input checked="" type="checkbox"/>

The above entries indicate that, if backup is successfully completed (i.e., not canceled by the operator), the return code for Job 1 is set to OK.

Job 2 (Posting)	
Directory Group	TRNSPP
Directory ID	POSTING
Check Code from Job	UTILS BACKUP
RC	In, OK – OK

The above entries indicate that the posting process cannot initiate until the system determines that backup has been successfully completed; that is, that the return code for Job 1 is set to OK.

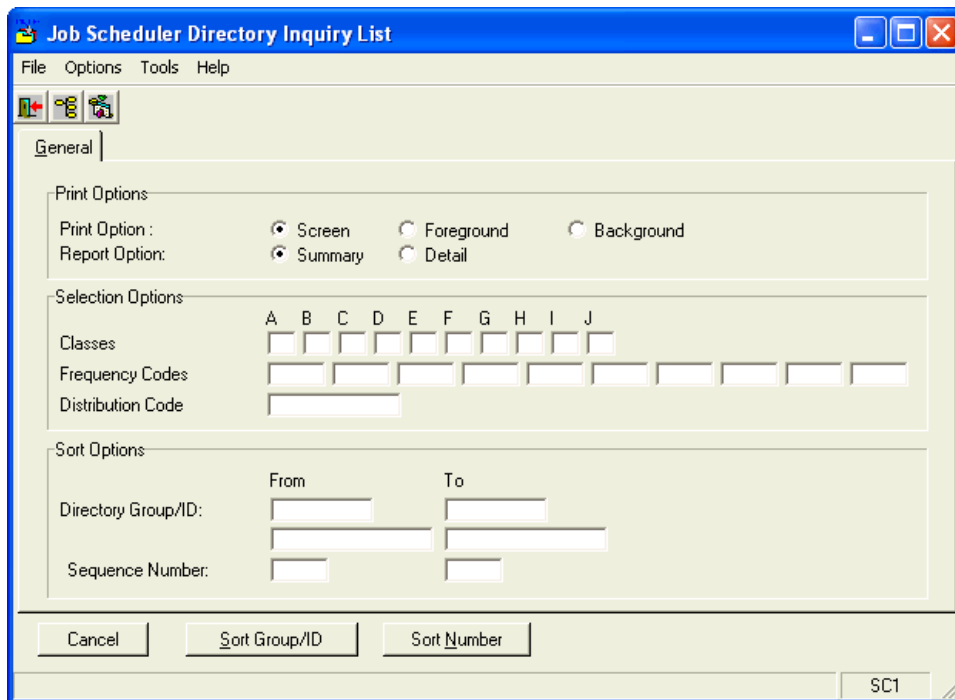
3.4 Job Scheduler Directory Inquiry/List

Introduction to Job Scheduler Directory Inquiry / List

This function is used to display online or print information concerning the jobs that have been set up in the Job Scheduler Directory established via the Job Scheduler Directory Maintenance function.

Job Scheduler Directory Inquiry / List Window (SC1)

The SC1 window displays when you select the Job Scheduler Directory Inquiry/List function.



Directory Inquiry/List Window (SC1)

Print Options Panel

Print Option

This is the processing option that produces the online display or the printed report. Valid options are:

Screen – This option enables you to review a list of the data online, as well as to display detail for each item on the list.

Foreground – This option produces a print file, processing the request in foreground. Use of this option restricts use of this window for other processing.

Background – This option produces a print file, processing the request in background. Use of this option allows you to continue to process other functions from this window.

Default: Screen.

Report Option

This field specifies the type of report to generate. Valid options are:

Summary – The report lists the following information for each job: sequence number, directory group, directory ID, status, distribution code, description, run program, run Library, volume, and comment.

Detail – The report lists the above information and several additional parameters for each job, including any messages or record selection criteria.

Default: Summary.

Selection Options Panel

The fields on this panel allow you to specify the data to access.

Classes / Frequency Codes / Distribution Code

These fields can be used to restrict the data accessed to particular jobs. To select jobs by class, enter an X in the appropriate box(es). To select jobs by frequency code or by distribution code, enter the desired code(s).

Sort Options Panel

The fields in this panel allow you to determine the sort option for the report or online display. They also allow you to restrict the data accessed.

Directory Group/ID

These fields specify the jobs to display in order by directory group and, within directory group, by directory ID. (If the sort option is Sort Group/ID, see below.) Two rows of boxes are shown to the right of this field. The first row is used to specify a range of directory groups. Leave the fields blank to display all directory groups. The second row is used to specify a range of directory IDs within a single directory group. Leave these fields blank to display all directory IDs.

Sequence Number

These fields specify the jobs to display in order by sequence number if the sort option is Sort Number, see below.) Enter a range of sequence numbers. (Leave blank to display all sequence numbers.)

Enter the appropriate information and select the desired sort option:

Button	Function
Sort Group/ID	Display the jobs in order by directory group, and within directory group, by directory ID.
Sort Number	Display the jobs in order by sequence number.

Job Scheduler Report Description

Print Options

The following sample reports are available for the inquiry/list function.

Batch Stream Directory Report (Summary)

```

*   COMPUTRON SOFTWARE, LLC.   ***** Wed May 02, 2007 10:42 AM *
*
*                               JOB SCHEDULER DIRECTORY SUMMARY REPORT
*
*----- Page 1 *

```

Seq#	Group	Job ID	ST	Distrib	Description	Run Prog	Run Lib	Run Vol	Comment
	APPURCH	PURCH1	Y		Purchase Trend Analysis Rpt	PURCRPB	CIAPRUN		
0	APVOUIL0	TEST	Y		Open Voucher Inquiry/List	OPENVILL	CIAPRUN		
0	FARPT	PROPL	Y		Property Listing Report	FAREPRT1	CIFARUN		
0	FARPT	DISPL	Y		DISPOSALS REPORT	FAREPRT1	CIFARUN		
0	FARPT	EXPR1	Y		DEPRECIATION EXPENSE REPORT	FAREPRT1	CIFARUN		
0	FARPT	SNAP	Y		Data Extraction for Above Rpts	FAREPRT1	CIFARUN		
0	COMMENT	PRELIMCMT	Y		THIS IS A MESSAGE ONLY JOB				
10	APREGPP	EDITPOST	Y		POSTING - EDIT ONLY	APREGPP1	CIAPRUN		
10	FAPRO	FAPROCESS	Y		Transaction Processing	TRANSUP1	CIFARUN		
10	APREGPP	APREGPSE	Y		POSTING - EDIT ONLY	APREGPP1	CIAPRUN		
20	APREGPP	UPDATEPOST	Y		Posting Update	APREGPP1	CIAPRUN		
40	APCKPP	PRELIMCK	Y		PRELIMINARY CHECK REGISTE	WRITICK0	CIAPRUN		
50	FABATREL	BATCHEDIT	Y		Batch Transaction Release	FABATRL1	CIFARUN		
55	APRECRPT	APRECON	Y		AP Reconciliation Report	APRECON1	CIAPRUN		
60	APRECRPT	CASHRECON	Y		AP CASH RECON REPORT	APRECON1	CIAPRUN		
75	AGEDCSH	AGEDCASH	Y		Aged Cash Requirements Rpt	AGEDCSH1	CIAPRUN		
75	FABATREL	BATCHREL	Y		Batch Transaction Release	FABATRL1	CIFARUN		
85	FAPOST	FAGLEPOST	Y		FA TO G/L EDIT POSTING	GLPOST1	CIFARUN		
95	FAPOST	FAGLPOST	Y		FA TO GENERAL LEDGER POSTING	GLPOST1	CIFARUN		

Job Scheduler Directory Report (Detail)

```

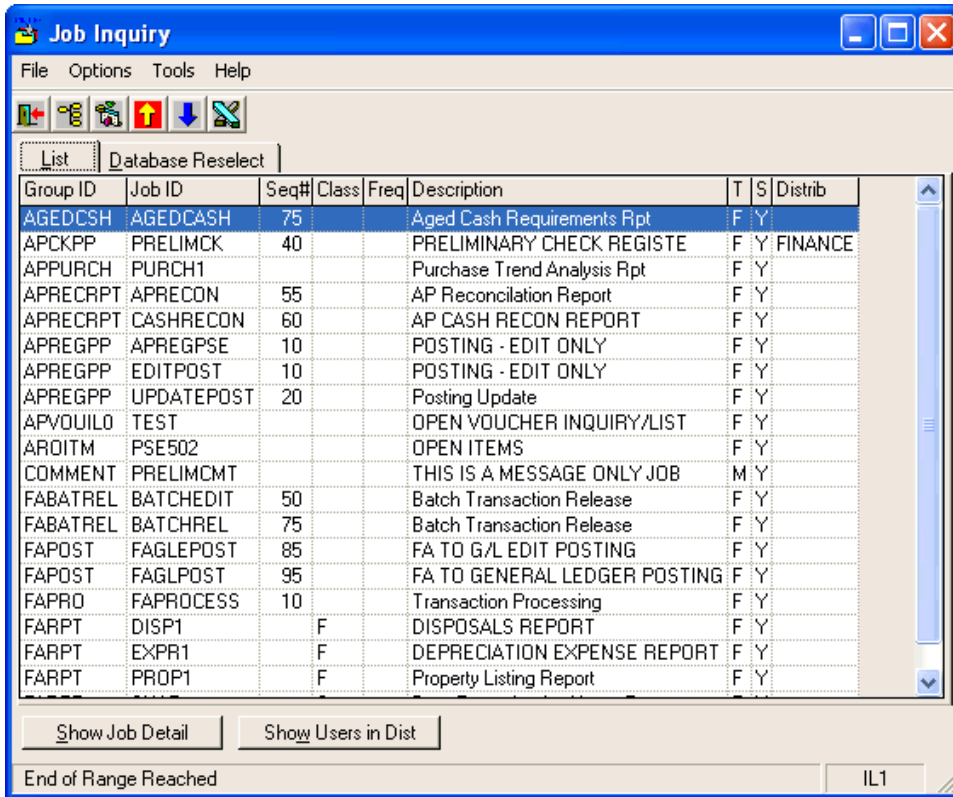
*   Computron Software, LLC.   ***** Wed May 02, 2007 10:42 AM *
*
*                               JOB SCHEDULER DIRECTORY DETAIL REPORT
*                               SORTED BY: DIRECTORY GROUP/ID
*
*----- Page 1 *

```

GROUP	ID	SEQ	DESCRIPTION / COMMENTS	EFFECTIVE/ST	OL/DISTRIB	FREQ	RUN PROGRAM	CLASSES	RUN MODE/EFFECTIVE	SL	LIB/VOLUME	PROMPT/TYPE
AGEDCSH	AGEDCASH	75	Aged Cash Requirements Rpt	Y					10/24/97 05/01/07	9	AGEDCSH1 CIAPRUN	BACKGROUND NONE
APCKPP	PRELIMCK	40	PRELIMINARY CHECK REGISTE	Y					10/27/97 05/01/07	9	WRITICK0 CIAPRUN	BACKGROUND START OTHER
APPURCH	PURCH1		Purchase Trend Analysis Rpt	Y					09/26/06	9	PURCRPB CIAPRUN	BACKGROUND NONE OTHER
APRECRPT	APRECON	55	AP Reconciliation Report	Y					05/01/07	9	APRECON1 CIAPRUN	BACKGROUND NONE OTHER
APRECRPT	CASHRECON	60	AP CASH RECON REPORT	Y					05/01/07	9	APRECON1 CIAPRUN	BACKGROUND NONE OTHER
APREGPP	APREGPSE	10	POSTING - EDIT ONLY	Y					05/01/07 05/01/07	9	APREGPP1 CIAPRUN	BACKGROUND NONE OTHER
APREGPP	EDITPOST	10	POSTING - EDIT ONLY	Y					10/23/97 05/01/07	9	APREGPP1 CIAPRUN	BACKGROUND NONE OTHER
APREGPP	UPDATEPOST	20	Posting Update	Y					10/23/97 05/01/07	9	APREGPP1 CIAPRUN	BACKGROUND NONE OTHER
APVOUIL0	TEST	0	Open Voucher Inquiry/List	Y					08/30/06 08/30/06	9	OPENVILL CIAPRUN	BACKGROUND NONE OTHER
COMMENT	PRELIMCMT	0	THIS IS A MESSAGE ONLY JOB	Y					05/02/07	0		MESSAGE START UTILITY
MESSAGES: THIS IS A MESSAGE FOR THE OPERATOR. THE JOB IS READY TO COMMENCE.												
FABATREL	BATCHEDIT	50	Batch Transaction Release	Y					10/24/97 05/01/07	9	FABATRL1 CIFARUN	BACKGROUND NONE OTHER
FABATREL	BATCHREL	75	Batch Transaction Release	Y					10/24/97 05/01/07	9	FABATRL1 CIFARUN	BACKGROUND NONE OTHER

Screen Option

When you request the Job Scheduler Directory I/L function to produce the information to Screen, the following IL1 window displays.

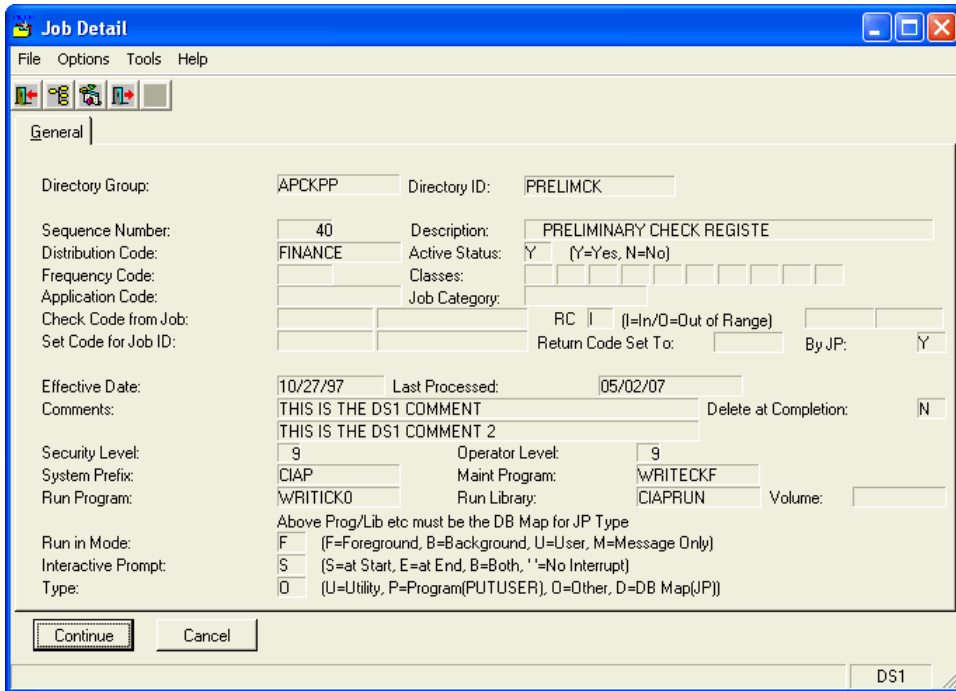


The screenshot shows a window titled "Job Inquiry" with a menu bar (File, Options, Tools, Help) and a toolbar. Below the toolbar are buttons for "List" and "Database Reselect". The main area contains a table with the following columns: Group ID, Job ID, Seq#, Class, Freq, Description, T, S, and Distrib. The table lists various jobs such as AGEDCSH, APCKPP, APPURCH, etc. At the bottom of the window, there are buttons for "Show Job Detail" and "Show Users in Dist", and a status bar showing "End of Range Reached" and "IL1".

Group ID	Job ID	Seq#	Class	Freq	Description	T	S	Distrib
AGEDCSH	AGEDCASH	75			Aged Cash Requirements Rpt	F	Y	
APCKPP	PRELIMCK	40			PRELIMINARY CHECK REGISTE	F	Y	FINANCE
APPURCH	PURCH1				Purchase Trend Analysis Rpt	F	Y	
APRECRPT	APRECON	55			AP Reconciliation Report	F	Y	
APRECRPT	CASHRECON	60			AP CASH RECON REPORT	F	Y	
APREGPP	APREGPSE	10			POSTING - EDIT ONLY	F	Y	
APREGPP	EDITPOST	10			POSTING - EDIT ONLY	F	Y	
APREGPP	UPDATEPOST	20			Posting Update	F	Y	
APVOUIL0	TEST				OPEN VOUCHER INQUIRY/LIST	F	Y	
ARDITM	PSE502				OPEN ITEMS	F	Y	
COMMENT	PRELIMCMT				THIS IS A MESSAGE ONLY JOB	M	Y	
FABATREL	BATCHEDIT	50			Batch Transaction Release	F	Y	
FABATREL	BATCHREL	75			Batch Transaction Release	F	Y	
FAPOST	FAGLEPOST	85			FA TO G/L EDIT POSTING	F	Y	
FAPOST	FAGLPOST	95			FA TO GENERAL LEDGER POSTING	F	Y	
FAPRO	FAPROCESS	10			Transaction Processing	F	Y	
FARPT	DISP1		F		DISPOSALS REPORT	F	Y	
FARPT	EXPR1		F		DEPRECIATION EXPENSE REPORT	F	Y	
FARPT	PROP1		F		Property Listing Report	F	Y	

Job Scheduler Directory Inquiry Window (IL1)

When you click the Show Job Detail button, the DS1 window displays the details of the highlighted job.



Job Detail

File Options Tools Help

General

Directory Group: APCKPP Directory ID: PRELIMCK

Sequence Number: 40 Description: PRELIMINARY CHECK REGISTE

Distribution Code: FINANCE Active Status: Y (Y=Yes, N=No)

Frequency Code: Classes:

Application Code: Job Category:

Check Code from Job: RC (I=In/O=Out of Range)

Set Code for Job ID: Return Code Set To: By JP: Y

Effective Date: 10/27/97 Last Processed: 05/02/07

Comments: THIS IS THE DS1 COMMENT Delete at Completion: N
THIS IS THE DS1 COMMENT 2

Security Level: 9 Operator Level: 9

System Prefix: CIAP Maint Program: WRITECKF

Run Program: WRITICK0 Run Library: CIAPRUN Volume:

Above Prog/Lib etc must be the DB Map for JP Type

Run in Mode: F (F=Foreground, B=Background, U=User, M=Message Only)

Interactive Prompt: S (S=at Start, E=at End, B=Both, ''=No Interrupt)

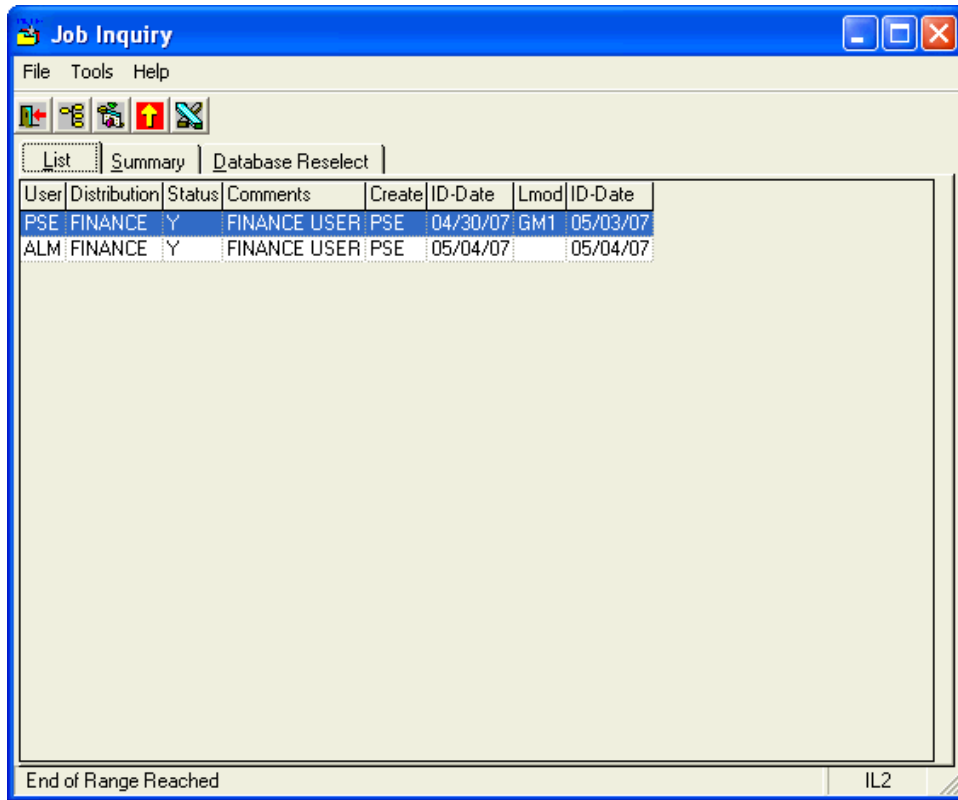
Type: O (U=Utility, P=Program(PUTUSER), O=Other, D=DB Map(JP))

Continue Cancel

DS1

Job Scheduler Directory Inquiry Window (DS1)

When you highlight a job with a distribution list and click the Show Users in Dist button, the IL2 window displays the list of users who are to receive copies of the print out of the job.



Job Scheduler Directory Inquiry Window (IL2)

3.5 User Code Maintenance

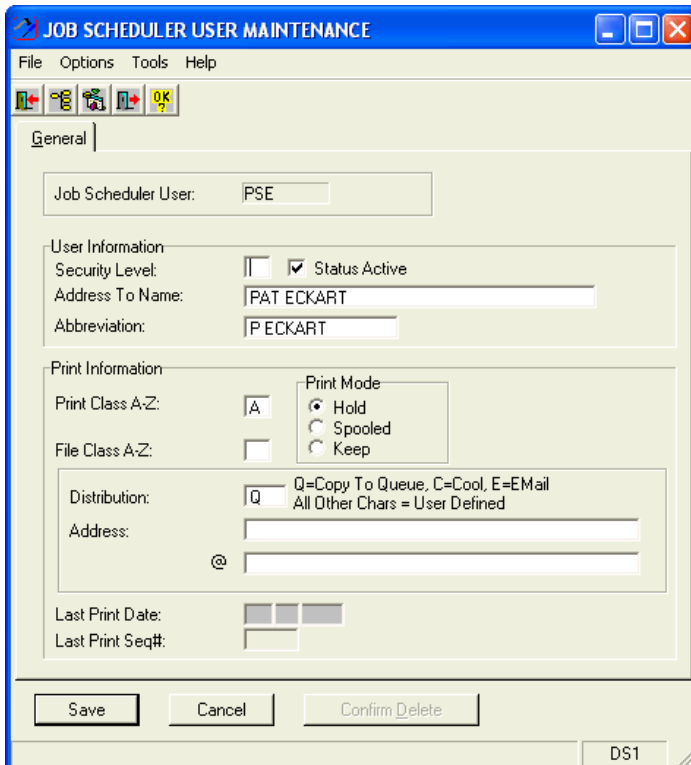
Introduction to User Code Maintenance

Each user targeted to receive copies of reports, as generated via Audit Processor, should be assigned a user code. The user code can be cross-referenced with one or more distribution codes through User Distribution List Maintenance. The cross-referencing via User Distribution List Maintenance determines which reports the user receives.

The Audit Processor generates copies of reports for the designated users, stores these reports in print files and assigns the print files to the appropriate libraries. The Audit Processor also produces the Audit report listing the users and the reports they are to receive.

User Maintenance Window (DS1)

From the User Maintenance Window (KS1), enter a value in the User field to maintain (Q-^{Mark}), and select the desired maintenance option, Add, Change, Delete, or Copy. The User Maintenance Window (DS1) displays.



Job Scheduler User Maintenance Window (DS1)

Security Level

Assign a security level to the user code record. This security level is used in conjunction with data established in the User file, to restrict use of this code. The associated record is available only to those users with an equal or higher security level. Valid entries are 0-9.

Status Active

Enter the status of the code to maintain. This field appears on Q-^{Mark} windows. This is a required field. Valid options are:

- This option indicates the code is active and can be used.
- This option indicates the code is inactive.

Default:

Address To Name

Enter the user's name, as it will appear on the Audit report. This field appears on Q-^{Mark} windows.

Abbreviation

Enter an abbreviation of up to ten alphanumeric characters for the Address To Name value. This abbreviation is used on reports when the Address To Name value is too long to print in its entirety. This field appears on Q-^{Mark} windows.

Print Class A-Z

Enter a letter A-Z to specify the print class to assign to the user's print files. Each printer is assigned certain print classes and will print only files in those classes.

Print classes are defined in a file called LPMAP. The LPMAP file acts as a table of different parameter settings for printer control codes. The LPMAP file is used in conjunction with the PRMAP file, which further describes the available printers, or a file called FORMS, which defines the control codes for formatting the printed text, e.g., font, characters per inch, lines per inch, etc.

Default: A.

File Class A-Z

Enter a letter (A-Z) to specify the file protection class to assign to the user's print files. These classes are defined by System Security Administrators and are unique for each installation.

Print Mode

This field determines how the system handles the user's print files. Valid options are:

H – Control of the print file is transferred to the print spooling system, which places an entry for the file on the print queue in a hold status pending user action.

S – Control of the print file is transferred to the print spooling system, which places an entry for the file on the print queue. The file is printed automatically on an appropriate printer as soon as the printer is available.

K – Print output is stored in a print file and assigned to a library on disk. However, control of the file is not passed to the print spooling system and no entry is placed on the print queue. Not currently implemented. (If selected, the field value defaults to H).

Default: H

Distribution

This field determines how the system distributes the user's print files. Valid options are:

Q – The print file is copied to the appropriate user print library.

C – The print file is passed to the COOL application to be indexed. Not currently implemented.

E – The print file is sent via electronic mail to the appropriate user (using the electronic mailing address specified in the Address field).

☞ *Note:* If you enter a value other than Q, C or E in this field, the system interprets this value as a pointer to a customized distribution script.

Address

Enter the user's Internet ID, if the Distribution field is set to E.

@

Enter the user's Internet address, if the Distribution field is set to E.

Last Print Date

This field displays the last date on which print files were generated for the user.

☞ *Note:* This field is updated by the system and cannot be modified.

Last Print Seq #

As print files are created, the system assigns Print Sequence Numbers. As a result, this field displays the last number used.

☞ *Note:* This field is updated by the system and cannot be modified.

3.6 User Distribution List Maintenance

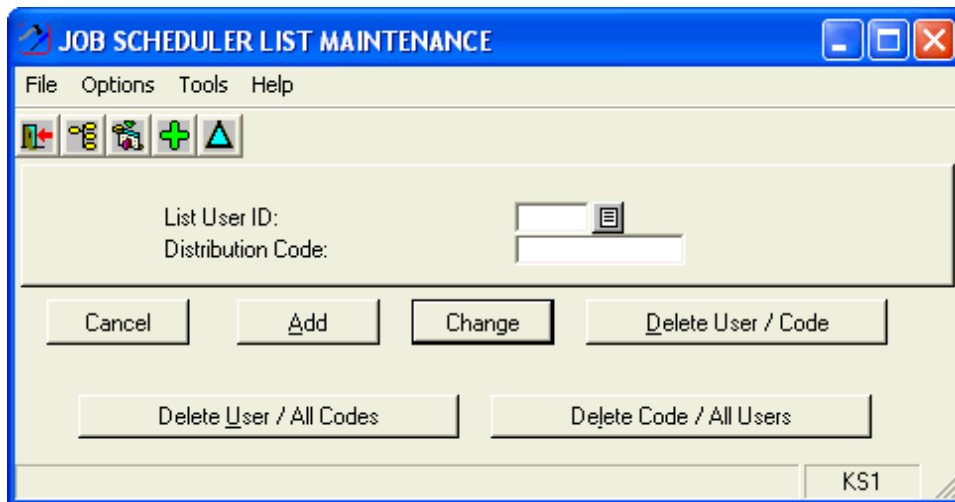
Introduction to User Distribution List Maintenance

Use this function to maintain the User Distribution List a file that cross-references distribution codes with the codes for users who are to receive copies of reports. Based on the information in this file, the Audit Processor automatically produces multiple copies of reports and distributes them to the appropriate users' print file libraries.

Note: The user codes and distribution codes that are cross-referenced must first be defined via the User Code Maintenance and Distribution Code Maintenance functions.

User Distribution List Maintenance Window (KS1)

The KS1 window displays when you select the User Distribution List Maintenance function.



User Distribution List Maintenance Window (KS1)

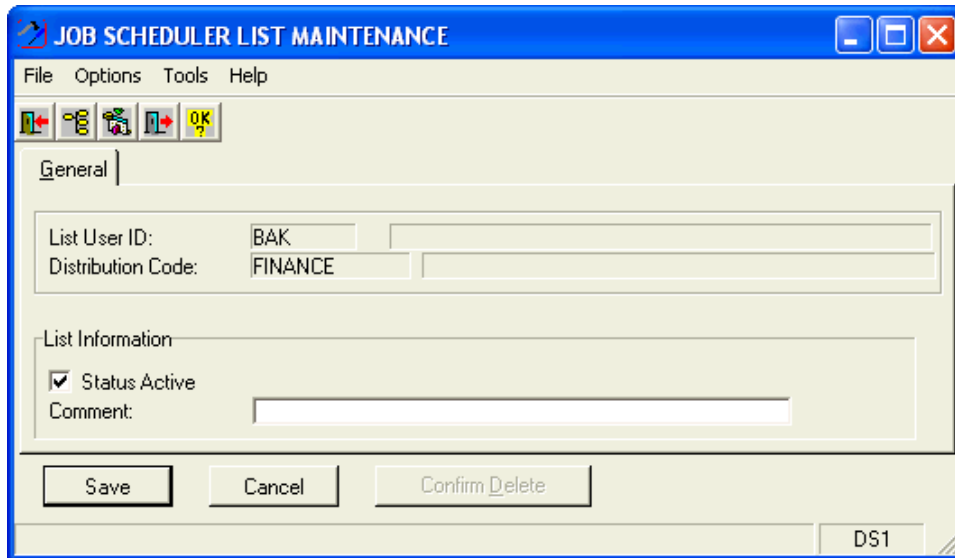
Button	Function
Delete User / Code	Delete a user code from a distribution code.
Delete User / All Codes	Delete a user code from all distribution codes. (Leave the Distribution Code field blank)
Delete Code / All Users	Delete a distribution code from all user codes. (Leave the List User ID field blank.)

From the User Distribution List Maintenance Window (KS1) enter the appropriate list user ID (Q-Mark), and distribution code to maintain. Then select the desired maintenance option, Add, Change, Delete/User Code, Delete User/All Codes or Delete Code/All Users.

The User Distribution List Maintenance Window (DS1) displays when you click the Add or Change button.

User Distribution List Maintenance Window (DS1)

The DS1 window displays when you click the Add or Change button on the KS1 window.



User Distribution List Maintenance Window (DS1)

Status Active

Enter the status of the User Distribution list entry. This field appears on Q-^{Mark} windows. This is a required field. Valid options are:

- This indicates the code is active. The Audit Processor uses this cross-reference entry when distributing reports.
- This indicates the code is inactive. The Audit Processor does not distribute reports.

Default:

Comments

Enter descriptive information regarding the cross-reference entry with up to thirty alphanumeric characters. These comments display on Q-^{Mark} windows, and on the User Distribution report.

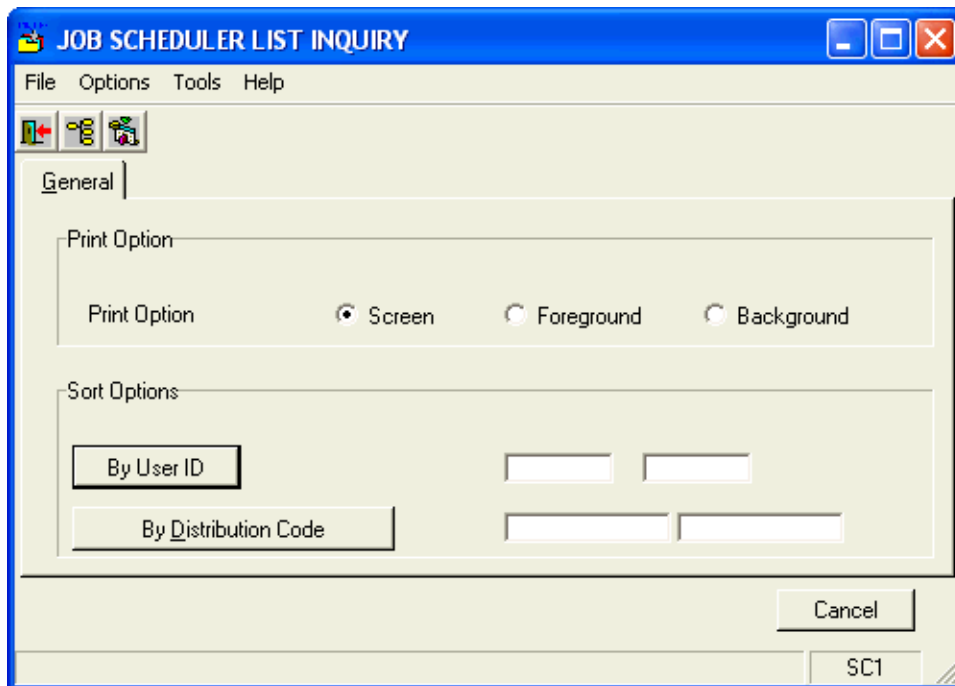
3.7 User Distribution List Inquiry / List

Introduction to User Distribution List Inquiry / List

This function is used to display a listing of the user codes within the distribution codes. This data can either be displayed online or spooled to a print file.

User Distribution List Inquiry Window (SC1)

The SC1 window displays when you select the User Distribution List I/L function.



User Distribution List Inquiry Window (SC1)

Print Option

This field indicates the processing option that produces either the online display or the printed report. Valid options are:

Screen – This option enables you to review a list of the data online, as well as to display detail for each item on the list.

Foreground – This option produces a print file, processing the request in foreground. Use of this option restricts use of this window for other processing.

Background – This option produces a print file, processing the request in background. Use of this option allows you to continue to process other functions from this window.

Default: Screen

Sort Options

By User ID

This specifies the cross-reference entries to display if the sort option is by user ID. Enter a range of user IDs. Leave these fields blank to display all user IDs.

By Distribution Code

This specifies the cross-reference entries to display if the sort option is by distribution code. Enter a range of distribution codes. Leave these fields blank to display all distribution codes.

Enter the appropriate information and select the desired sort option:

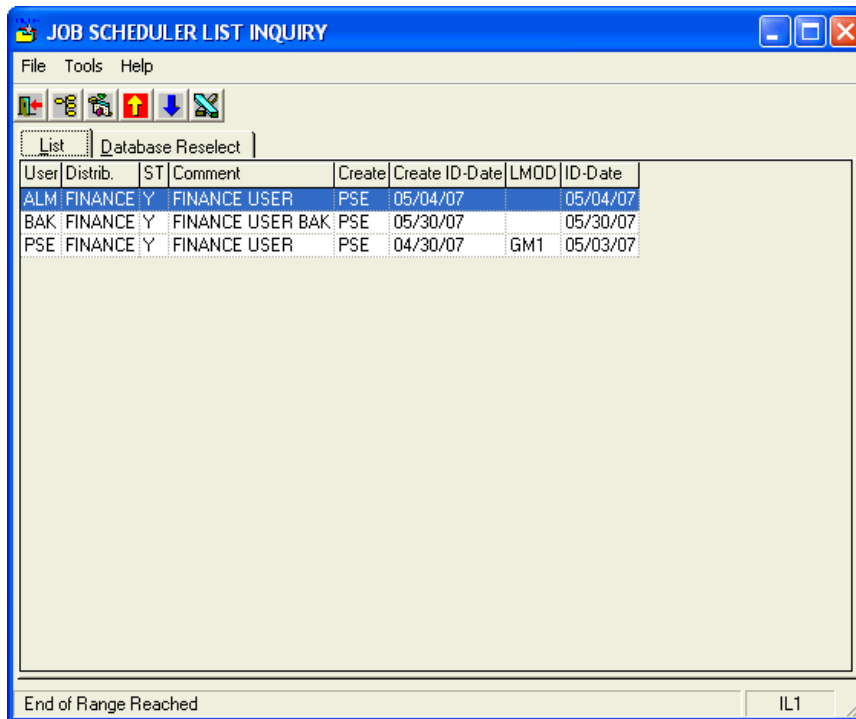
Button	Function
By User ID	Display the cross-reference entries in order by user ID.
By Distribution Code	Display the cross-reference entries in order by distribution code.

User Distribution List Report Options

Screen Option

When you select the Screen option, the IL1 window displays with all the users on it. The same IL1 window displays regardless of which option, by User ID or By Distribution Code, you request.

User Distribution List Window (IL1)



The screenshot shows a window titled "JOB SCHEDULER LIST INQUIRY" with a menu bar (File, Tools, Help) and a toolbar. Below the toolbar are two buttons: "List" and "Database Reselect". The main area contains a table with the following data:

User	Distrib.	ST	Comment	Create	Create ID-Date	LMOD	ID-Date
ALM	FINANCE	Y	FINANCE USER	PSE	05/04/07		05/04/07
BAK	FINANCE	Y	FINANCE USER BAK	PSE	05/30/07		05/30/07
PSE	FINANCE	Y	FINANCE USER	PSE	04/30/07	GM1	05/03/07

At the bottom of the window, it says "End of Range Reached" and "IL1".

User Distribution List Window (IL1)

Print Options

The following sample report is available from the inquiry/list function:

User Distribution Report

```

*   COMPUTRON SOFTWARE, LLC.   ***** Wed May 02, 2007 11:55 AM *
*                               ---- USER DISTRIBUTION LIST REPORT ----
*                               ***** Page      1 *

```

USER ID	DISTRIB	STATUS	COMMENT	CREATED ID	CREATED DATE	LMOD ID	LMOD DATE
PSE	FINANCE	Y	FINANCE USER	PSE	Apr 30, 07		Apr 30, 07

3.8 Job Processor

Introduction to Job Processor

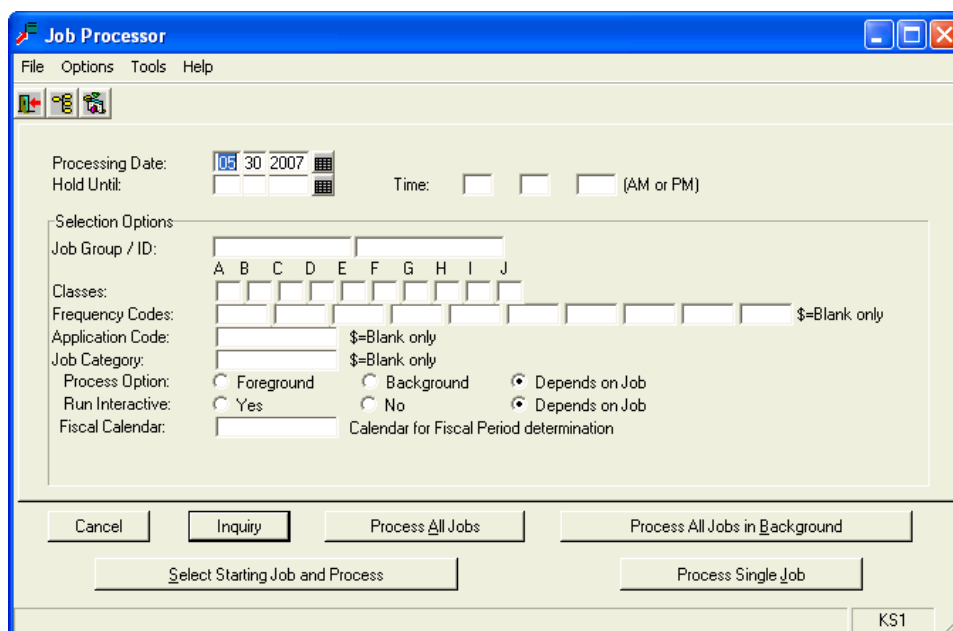
Use this function to control how jobs are run through the job stream. It allows you to process individual jobs or an entire stream of jobs. If for any reason the job stream is canceled mid-run, you can use this function to select any job from the stream and resume processing. The Job Process report is automatically produced when the Job Processor is run. This report details the job status (completed, canceled or skipped by the operator). In addition, for jobs that are run using Job Processor-Prepared Computron programs, a record is written to the Audit file for future use by the Audit Processor.

Additional Considerations

- ◆ To produce multiple copies of reports and distribute them to the appropriate user print libraries, run the Audit Processor function after running this function or set it as the last job to run in the Job Scheduler Directory. (The Audit Processor may also be put into a procedure.)
- ◆ If you are using return codes, clear the return codes following each run of the Job Processor. This can be accomplished via Return Code Inquiry/List.

Job Processor Window (KS1)

The KS1 window displays when you select the Job Processor function:



Job Processor Window (KS1)

Processing Date

Enter the “as of date” to use for processing.

Validation:

This date is compared to the effective date for each job, to determine whether the system should process the job. If the effective date is the same as or earlier than the processing date, the job is run.

If selecting jobs by frequency code (system options), the program checks this code to determine if it coincides with the processing date. For instance, if the frequency code is “#23” (i.e., 23rd day of the month) and the processing date is October 23rd, the job is run.

The processing date is also used to convert generic global date codes into specific dates. Thus, if the generic global date code is EOM (i.e., last day of the current month) and the processing date is October 23rd, the date code is interpreted as October 31st.

See Section 3.3, “Job Scheduler Directory Maintenance” for additional information.

Hold Until Time

Use this field to prevent the system from processing the job stream until a specific time and date has been reached.

☞ *Note:* This field is most effectively used when the job stream is to be submitted to background for processing. If the job stream is set to run at a particular time and submitted to foreground, the terminal is unavailable for other processing.

Example: To start the job stream as of 11:00 on May 29th, enter:

Job Processor Specification

Hold Until	05/29/2007
Time	11:00 PM

To hold the same job stream for processing until 1:00 AM on May 30th, enter:

Job Processor Specification

Hold Until	05/30/2007
Time	01:00 AM

☞ *Note:* Hours are entered using 1:00 to 12:00 combined with AM and PM designations.

Job Group/ID

Enter a value to identify the job stream.

Classes

You can select the jobs to process based on their assigned classes. Enter the character assigned to the job(s) you wish to select in the box(es) corresponding to the appropriate class (A–J).

☞ *Note:* See Section 3.3, “Job Scheduler Directory Maintenance” for additional information.

Frequency Codes (Q-^{Mark})

Select the jobs to process based on frequency code. You can specify up to 10 different codes. To select using the system options, enter “#” or “*”. The system evaluates all jobs with frequency codes beginning “#” or “*” to determine whether they should be processed (see Processing Date field). Jobs can also be selected based on the user-defined codes established through Frequency Code Maintenance. Leave the fields blank to select without reference to frequency code.

☞ *Note:* Enter “\$” to select only those jobs for which the frequency code field is blank. If all job frequency code fields are blank, all jobs are selected. If you enter “#” or “*”, the system selects the jobs appropriate for the date, in addition to those with blank frequency codes.

Any jobs with a blank frequency code are always selected.

Application Code Job Category (Q-^{Mark})

Additional selection criteria can be specified on the basis of application and job category codes. Enter the desired code(s) or leave the fields blank to select without reference to application or job category codes. In addition, enter “\$” to select only those jobs for which the application or job category code fields are blank.

Process Option

This is the processing option. Valid options are:

Foreground – Submit the job stream to foreground. The jobs in the stream are processed as indicated in the Run Mode field through the Job Scheduler Directory Maintenance function if Foreground or Background was selected. Those jobs in the job stream for which the Run Mode field was set to User are submitted to foreground for processing.

Background – Submit the job stream to background. The jobs in the stream are processed as indicated in the Run Mode field through the Job Scheduler Directory Maintenance function if Foreground or Background was selected. Those jobs in the job stream for which the Run Mode field was set to User are submitted to background for processing.

Depends on Job – The jobs in the stream are processed as indicated in the Run Mode field through the Job Scheduler Directory Maintenance function if Foreground or Background was selected. Those jobs in the job stream for which the Run Mode field was set to User are submitted to foreground for processing.

Default: Depends on Job

Run Interactive

This field displays a special Interactive Prompt Window at the start and/or end of the job. This window can include a message to the operator, general information about the job or prompts that allow the user to continue or cancel processing.

☞ *Note:* Regardless of the value you enter in this field, the Interactive Prompt Window always appears following the cancellation of a job, informing the operator that the job “was aborted.”

Valid options are:

Yes – Display an interactive prompt at the start of each job, whether or not the Job Scheduler Directory provides for an interactive prompt for the job in question.

No – Do not display the interactive prompt unless the job is a “Message Only” job or has been canceled by the operator. If the Job Scheduler Directory provides for interactive prompts, they are suppressed.

Depends on Job – The Job Scheduler Directory determines whether the interactive prompts are displayed for a particular job.

Default: Depends on Job

☞ *Note:* See Section 3.3, “Job Scheduler Directory Maintenance” for additional information.

Fiscal Calendar

Enter the fiscal calendar to associate with this job stream.

The following five processing options are available:

Button	Function
Inquiry	Select this option to inquire about the jobs that have been set up in the Job Scheduler Directory before initiating processing. This option can also be used to select and process an individual job. Your inquiry can encompass all jobs or be subject to the selection criteria specified in the Selection Options section of the Job Processor Window (KS1).
Process All Jobs	Select this option to initiate processing of an entire stream of jobs. All jobs in the Job Scheduler Directory are processed in foreground, subject to the selection criteria specified in the Selection Options section of the Job Processor Window (KS1).
Process All Jobs in Background	Select this option to initiate processing of an entire stream of jobs. All jobs in the Job Scheduler Directory are processed in background, subject to the selection criteria specified in the Selection Options section of the Job Processor Window (KS1).
Select Starting Job and Process	Select this option to initiate processing, beginning with a selected job within the stream. If processing of the job stream was initiated previously and then canceled, this option can be used to restart processing. The selected job and all jobs with higher sequence numbers are processed subject to the selection criteria specified in the Selection Options section of the Job Processor Window (KS1).
Process Single Job	Select this option to process an individual job in a stream. The selected job is processed subject to the selection criteria specified in the Selection Option section of the Job Processor Window (KS1).

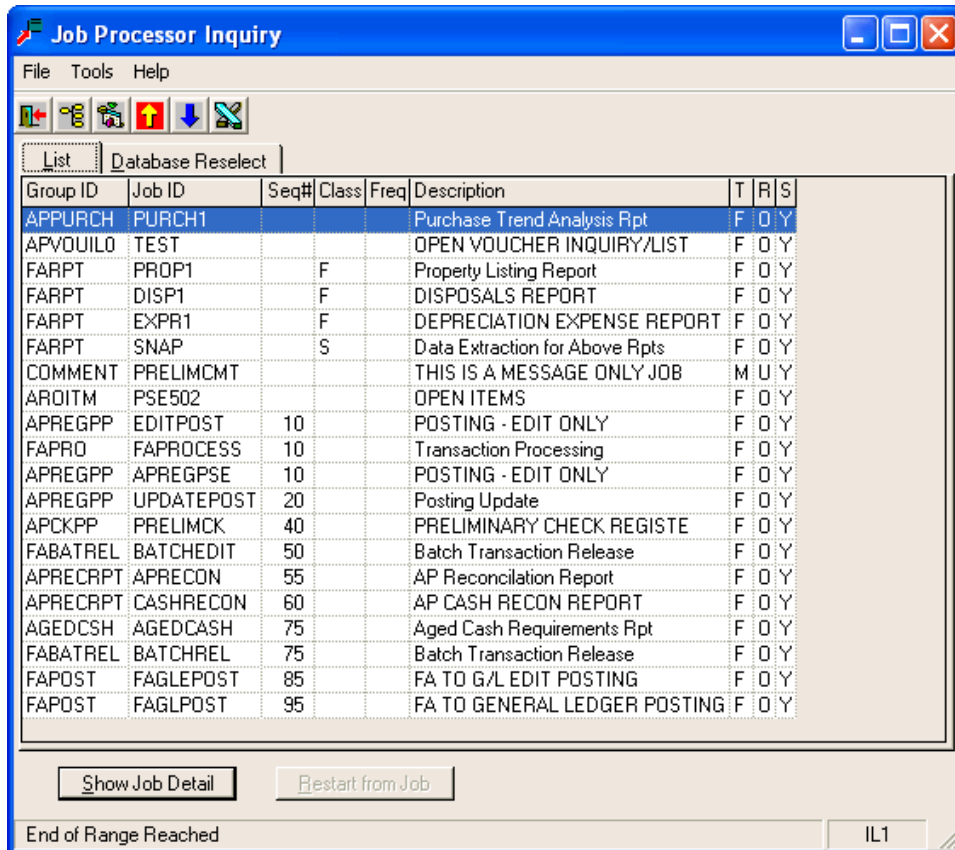
☞ *Note:* In all the above cases, the jobs are run in sequence number order.

If you select the Process All Jobs button, or the Process All Jobs in Background button, the Job Processor begins processing immediately.

When you select the Process Single Job button, the Job Processor initiates processing immediately for this job only.

Job Processor Inquiry Window (IL1)

The IL1 window displays when you select the Inquiry button:



Group ID	Job ID	Seq#	Class	Freq	Description	T	R	S
APPURCH	PURCH1				Purchase Trend Analysis Rpt	F	O	Y
APVOUILO	TEST				OPEN VOUCHER INQUIRY/LIST	F	O	Y
FARPT	PROP1		F		Property Listing Report	F	O	Y
FARPT	DISP1		F		DISPOSALS REPORT	F	O	Y
FARPT	EXPR1		F		DEPRECIATION EXPENSE REPORT	F	O	Y
FARPT	SNAP		S		Data Extraction for Above Rpts	F	O	Y
COMMENT	PRELIMCMT				THIS IS A MESSAGE ONLY JOB	M	U	Y
AROITM	PSE502				OPEN ITEMS	F	O	Y
APREGPP	EDITPOST	10			POSTING - EDIT ONLY	F	O	Y
FAPRO	FAPROCESS	10			Transaction Processing	F	O	Y
APREGPP	APREGPSE	10			POSTING - EDIT ONLY	F	O	Y
APREGPP	UPDATEPOST	20			Posting Update	F	O	Y
APCKPP	PRELIMCK	40			PRELIMINARY CHECK REGISTE	F	O	Y
FABATREL	BATCHEDIT	50			Batch Transaction Release	F	O	Y
APRECRPT	APRECON	55			AP Reconciliation Report	F	O	Y
APRECRPT	CASHRECON	60			AP CASH RECON REPORT	F	O	Y
AGEDCSH	AGEDCASH	75			Aged Cash Requirements Rpt	F	O	Y
FABATREL	BATCHREL	75			Batch Transaction Release	F	O	Y
FAPOST	FAGLEPOST	85			FA TO G/L EDIT POSTING	F	O	Y
FAPOST	FAGLPOST	95			FA TO GENERAL LEDGER POSTING	F	O	Y

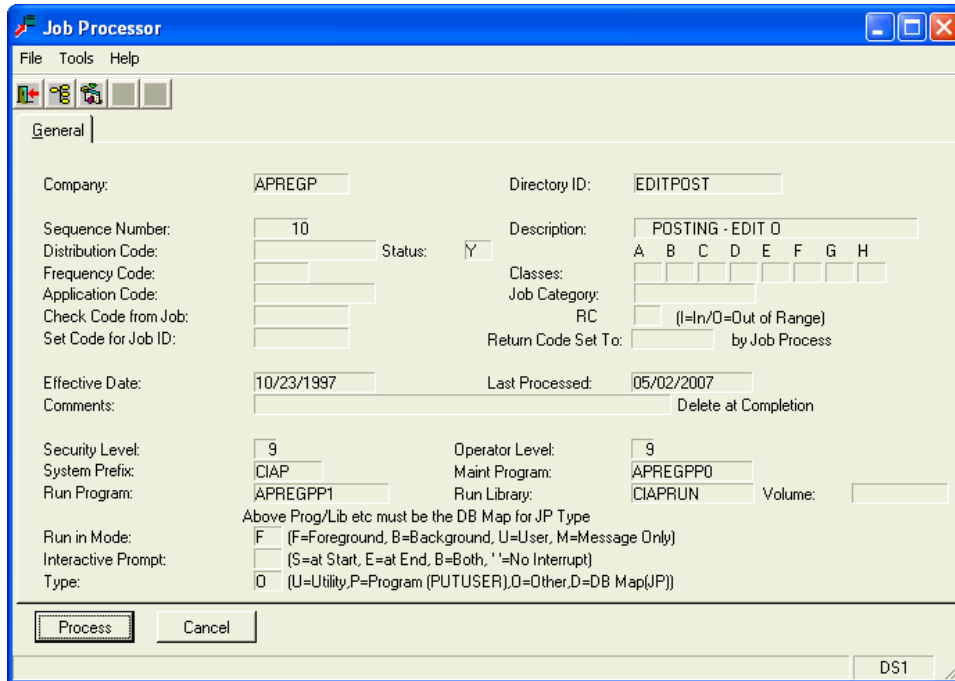
Job Processor Inquiry Window (IL1)

This window displays an ordered list (by sequence number) of jobs set up in the Job Scheduler Directory. From this window, review the original information entered for a particular job by positioning the cursor on the line and selecting Enter or the Select Item icon.

When you select the Select Starting Job and Process button, the Job Processor Inquiry Window (IL1) displays an additional button — the Restart from Job button. To initiate processing from a specific job, position the cursor on the job and select Enter or the Select Item icon. The selected job and all jobs with higher sequence numbers are processed, subject to the selection criteria specified earlier.

Job Processor Window (DS1)

The DS1 window displays when the Show Job Details button is clicked on the IL1 window.



The screenshot shows the 'Job Processor' window with the following fields and values:

- Company: APREGP
- Directory ID: EDITPOST
- Sequence Number: 10
- Description: POSTING - EDIT 0
- Distribution Code: (empty) Status: Y
- Frequency Code: (empty)
- Classes: A B C D E F G H
- Application Code: (empty)
- Job Category: (empty)
- Check Code from Job: (empty)
- RC: (empty) (I=In/O=Out of Range)
- Set Code for Job ID: (empty)
- Return Code Set To: (empty) by Job Process
- Effective Date: 10/23/1997
- Last Processed: 05/02/2007
- Comments: (empty) Delete at Completion
- Security Level: 9
- Operator Level: 9
- System Prefix: CIAP
- Maint Program: APREGPP0
- Run Program: APREGPP1
- Run Library: CIAPRUN
- Volume: (empty)

Below the fields, there is a note: "Above Prog/Lib etc must be the DB Map for JP Type".

Run in Mode: F (F=Foreground, B=Background, U=User, M=Message Only)

Interactive Prompt: (empty) (S=at Start, E=at End, B=Both, ''=No Interrupt)

Type: 0 (U=Utility, P=Program (PUTUSER), O=Other, D=DB Map(JP))

Buttons: Process, Cancel

Bottom right corner: DS1

Job Processor Window (DS1)

Processing Description

As the system initiates processing for each job in the stream, the system displays a message indicating the name of the program currently running. If a job is canceled by the operator, processing resumes with the next job in the stream.

Processing can be interrupted from time to time by the appearance of the Interactive Prompt window (see Run Interactive field). The operator is prompted to continue (by selecting the Continue option or Enter), to skip the next job (by selecting the Skip option) or to cancel any further processing for the stream (by selecting the Cancel option).

Note: The Cancel Job Stream button cancels the entire job stream, not just the current job.

When the job stream is processed, the Batch Stream Job Processor produces the Batch Stream Job Process report. This report shows the following information for each job selected for processing: sequence number, directory group, directory ID, distribution code, description, run program, run library, run volume, and start and end time. The report also indicates under the heading "Comments" whether a job was canceled or skipped.

Job Processor Report Description

The following sample report is available when the Job Processor is complete:

```

Job Processor Report
*   Computron Software, LLC   ***** Wed May 02, 2007  1:57 PM *
*                               --- JOB PROCESSOR REPORT---
*
*                               ***** Page      3 *
*****
PROCESS DATE: 05/02/07  CLASSES:          FREQUENCY CODES:          INTERACTIVE: D
APPLICATION CODE:      JOB CATEGORY:
SEQ# GROUP   JOB ID  DISTRIB   DESCRIPTION   PROGRAM  LIBRARY  VOLUME  START  END TIME  COMMENT  LRC#
-----
TOTAL      1  JOBS PROCESSED
TOTAL      1  RC 05  PROCESS COMPLETE BUT NO RECORDS FOUND FOR INDICATED CRITERIA

          --- JOB PROCESSOR ---          KS1
Enter Indicated Data and Press (RETURN)

Processing Date: 05 / 02 / 20 07 01/01/1901 = Today's Date
                                01/02/1901 = Look up Code TOD
Hold Until ----: / /          Time: : (AM or PM)

PF-1 = Inquiry
PF-2 = Process All Jobs / (PF-18) = in Background
PF-3 = Select Starting Job and Process
PF-4 = Process Single Job
PF-16 = EXIT

Selection Options:
Job Group / ID      A B C D E F G H I J
Classes
Frequency Codes
Application Code    $=Blank only  Job Category    $=Blank only
Ground             D (F)oreground, (B)ackground, (D)epends on Job
Run Interactive    D (Y)es, (N)o, (D)epends on Job
Fiscal Calendar   Calendar for Fiscal Period determination
UV v: 10.00.00 Copyright 2007 Computron Software, LLC. All Rights Reserved.

```

3.9 Audit Processor

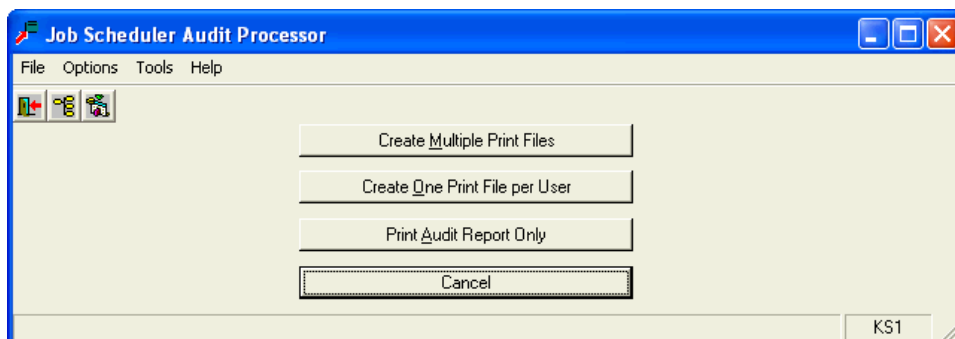
Introduction to the Audit Processor

This function can only be used after the Job Processor has processed jobs. For all jobs using Job Processor-Prepared Computron programs, the Job Processor writes a record to the Audit file. The Job Processor uses the information stored in the Audit file to create multiple copies of the reports generated by the job stream and to distribute these reports to the appropriate user print libraries.

Note: The Audit Processor function can be entered, via the Job Scheduler Directory Maintenance, as the last job in the stream (Sequence Number 9999). This function then runs automatically each time the Job Processor is run.

Audit Processor Window (KS1)

The KS1 window displays when you select the Audit Processor function:



Audit Processor Window (KS1)

Button	Function
Create Multiple Print Files	One print file is created for each report and each user on the distribution list for the report. The print files are distributed to the appropriate user print directories. Thus, each user print directory contains multiple files, one for each report that the user is to receive.
Create One Print File per User	All reports for a given user are stored in a single print file. The print files are then distributed to the appropriate user print directories.
Print Audit Report Only	Create only the bsau report without creating any reports for users.
Cancel	Return to the menu.

Audit Processor Report Description

The following sample report is available from the Audit Processor when you use any of the three report producing options.

```

Audit Report
* ..... Tue July 022, 2007 5:31 PM *
*
*           SPOOL AUDIT REPORT
*
* ..... Page 1 *
USER: ETT           SPOOL USER ID:           SPOOL USER ADDRESS: cettprt
----- DIRECTORY -----
NUM  SPOOL  GROUP  CODE  DISTR  ID  PROGRAM  PRFX  DATE  START  END  T  NAME  - SPOOLED  FILE -
-----
DISTRIB1 RPT1000B CIGL 05/21/07 12:16 12:16 F /ctron_3.1r/exec/spool/ccprt/rpt10001
1
PAGES
QUEUE GL1000 TEST
```

The Audit report details the distribution list for each report generated by the Job Processor. See Section 1.2, “User File Maintenance,” for additional information on the user data that displays on this report.

In addition, the Audit report lists the jobs that were run pertaining to each user. See Section 3.8, “Job Processor,” for additional information on the job data that displays on this report.

3.10 Return Code Inquiry/List

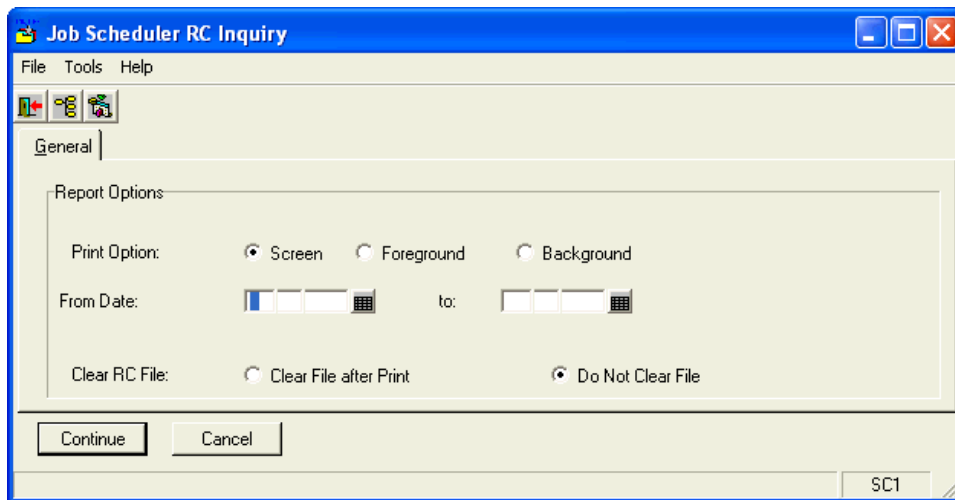
Introduction to Return Code Inquiry/List

Return codes are codes that are used to make one job dependent on the successful completion of another job. If a job is run from the Job Processor and is completed successfully, the return code is set to the value specified in the RC Set To Field of the Job Scheduler Directory, and written to the Return Code file. See Section 3.3, “Job Scheduler Directory Maintenance,” for additional information on the use of return codes,

This function is used to display online or print a list of return codes (established via Job Scheduler Directory Maintenance) that have been set following the completion (or cancellation) of jobs run from the Job Processor. This function is also used to clear the Return Code file, in preparation for a new run of the Job Processor.

Return Code Inquiry / List Window (SC1)

The SC1 window displays when you select the Return Code Inquiry/List function:



Return Code Inquiry/List Window (SC1)

Print Option

This is the processing option that produces either the online inquiry or the printed report. Valid options are:

Screen – This enables you to review a list of the data online, as well as to display detail for each item on the list.

Foreground – This option produces a print file, processing the request in foreground. Use of this option restricts use of this window for other processing.

Background – This option produces a print file, processing the request in background. Use of this option allows you to continue to process other functions from this window.

Default: Screen

From /To Date

This field allows you to select the information to be included in the report based on the dates on which the Return Codes were set. Enter a range of dates or leave blank to include all dates.

Clear RC File

This field allows you to indicate whether you wish to clear the Return Code file, following report or online inquiry generation. Valid options are:

Clear File after Print – Clear the Return Code file following completion of the inquiry or print file generation.

Do Not Clear File – Do not clear the Return Code file following completion of the inquiry or print file generation.

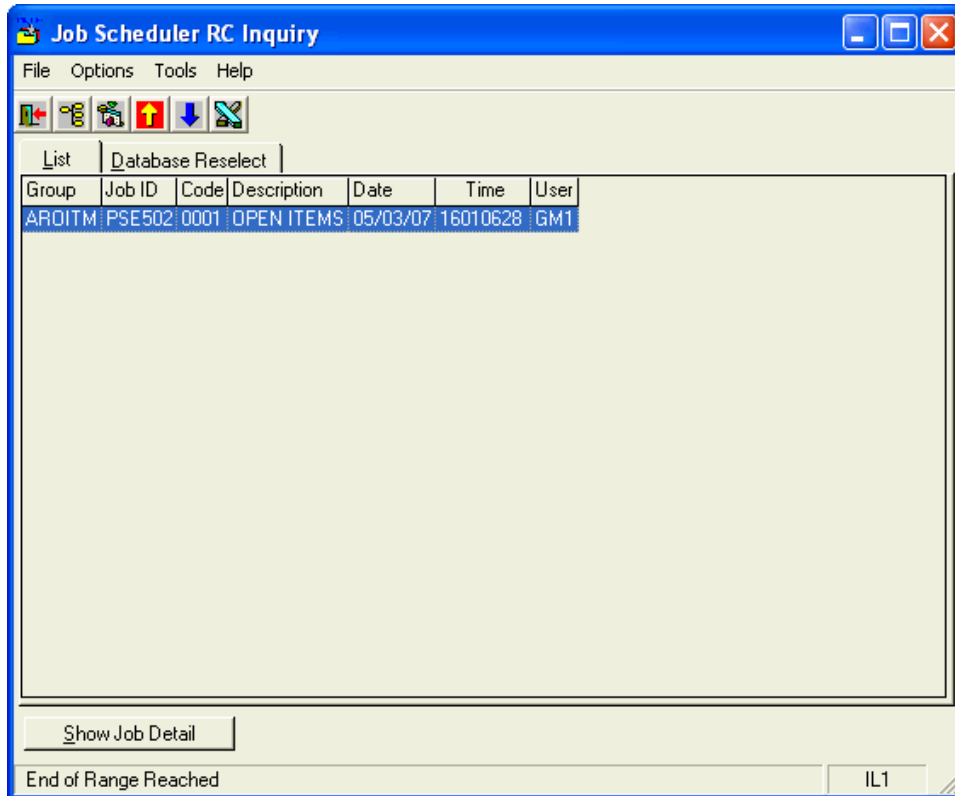
Default: Do Not Clear File

☞ *Note:* It is recommended that the return codes are cleared following each run of the Job Processor. Codes that are set when jobs are completed successfully will remain set to these values until cleared.

Return Codes Report Description

Return Codes Report Screen Option

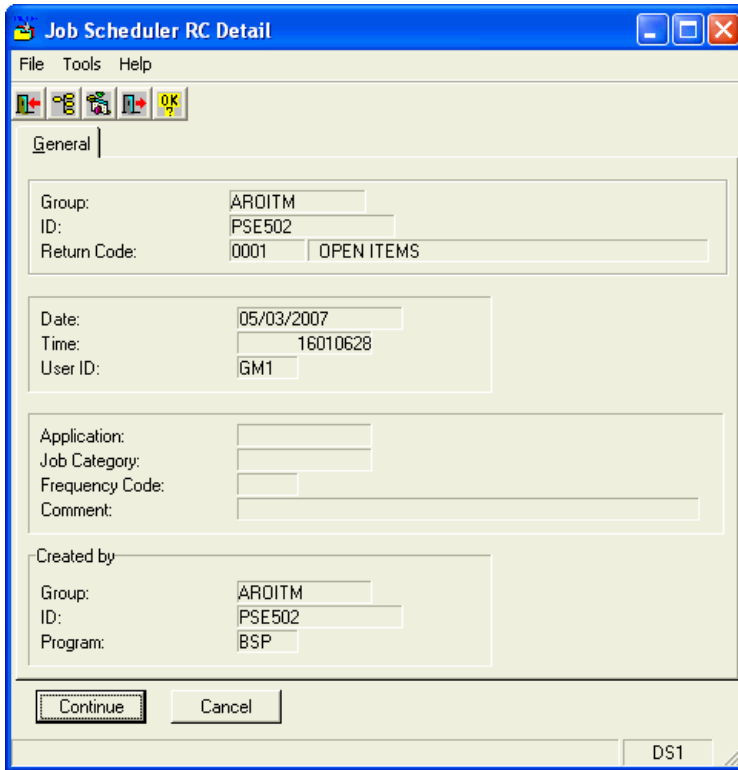
When the Screen option is selected, an inquiry window displays.



Return Code Inquiry/List Window (IL1)

To display the details of a listed code, you can either:

- ◆ highlight it and click the Choose toolbar button.
- ◆ double-click on the desired return code record.



Return Code Inquiry/List Window (DS1)

The inquiry detail window displays.

Return Codes Report Print Option

The following sample report is available from the Return Codes List:

```

Return Code Listing
*   COMPUTRON SOFTWARE, LLC.   ***** Tue July 02, 2007  5:29 PM *
*
*                               RETURN CODE REPORT
*
*----- Page 1 *
GROUP  ID      RETURN          DATE   TIME   USER  ID  GROUP  CREATED BY  ID      PROGRAM  COMMENT
CODE DESCRIPTION
-----
*** REPORT COMPLETED WITH      0 RECORDS PRINTED ***
--- BATCH STREAM RC INQUIRY ---
Enter Indicated Data and Press (RETURN)
Print Option : F (S=Screen, F=Foreground, B=Background)
From Date   :           to:           /           /
Clear RC File: NO (YES=Clear File after Print, or NO)
UV v: 10.00.00 Copyright 2007 Computron Software, LLC. All Rights Reserved.
    
```

3.11 Menu System Integration

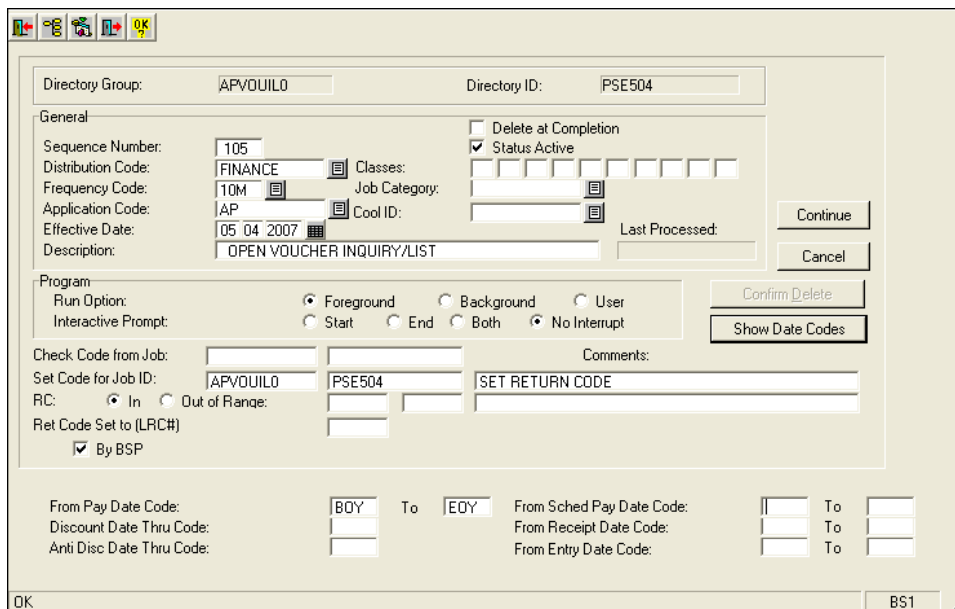
Introduction to Menu System Integration

You can integrate the Job Processing module with Computron's menu system via two methods:

- ◆ modify Computron Menus;
- ◆ create Job Processor Only Menus.

In both cases, menu integration has one basic purpose: to allow you to establish the Job Scheduler Directory parameters for Computron functions. This can also be accomplished from the Job Scheduler Directory Maintenance function. However, it may be preferable to allow individual users to access only those Computron applications with which they are familiar, rather than all functions within the system.

The system displays a BS1 window similar to the following when you select a menu integrated with Job Processing:



The screenshot shows a window titled "Job Scheduler Directory Window (BS1)". It contains several sections of configuration options:

- Directory Group:** APV0U1L0
- Directory ID:** PSE504
- General:**
 - Sequence Number: 105
 - Distribution Code: FINANCE
 - Frequency Code: 10M
 - Application Code: AP
 - Effective Date: 05 04 2007
 - Description: OPEN VOUCHER INQUIRY/LIST
 - Classes: (empty)
 - Job Category: (empty)
 - Cool ID: (empty)
 - Delete at Completion:
 - Status Active:
 - Last Processed: (empty)
- Program:**
 - Run Option: Foreground, Background, User
 - Interactive Prompt: Start, End, Both, No Interrupt
- Check Code from Job:** (empty)
- Set Code for Job ID:** APV0U1L0 PSE504
- RC:** In, Out of Range
- Ret Code Set to (LRCH#):** By BSP
- Comments:** SET RETURN CODE
- Date Codes:**
 - From Pay Date Code: BOY To EOY
 - From Sched Pay Date Code: (empty) To (empty)
 - From Receipt Date Code: (empty) To (empty)
 - From Entry Date Code: (empty) To (empty)

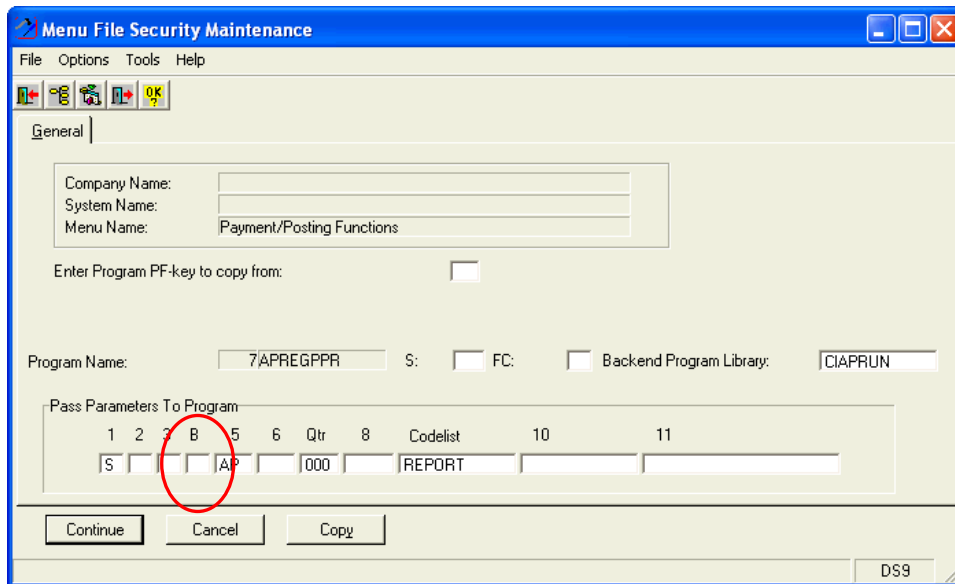
Buttons include "Continue", "Cancel", "Confirm Delete", and "Show Date Codes". The window title bar shows "OK" and "BS1".

Job Scheduler Directory Window (BS1)

The remaining windows are determined by the function selected. For Computron functions these windows correspond to those windows used to enter selection and sort criteria for the normal (non-job stream) option. See Section 3.3, "Job Scheduler Directory Maintenance," for additional information.

Modified Computron Menus

This approach enables you to restrict specific application functions to running via Job Processing only. For instance, assume that certain reports require a great deal of system resource and it is deemed most efficient to run them overnight. You can modify the standard Computron menus to present the Job Scheduler Directory Maintenance windows when that report function is selected from the menu. The following illustrates the type of entry required.



Menu Definition Window (DS9)

B

This field is used to indicate that the function is processed as part of Job Processing. Valid entries are:

M – This value indicates that the user is required to make entries on the Job Scheduler Maintenance Window (BS1) before the function can run. This window prompts the user for the information the Job Processor uses to run the function, e.g., the selection criteria to select records for reporting).

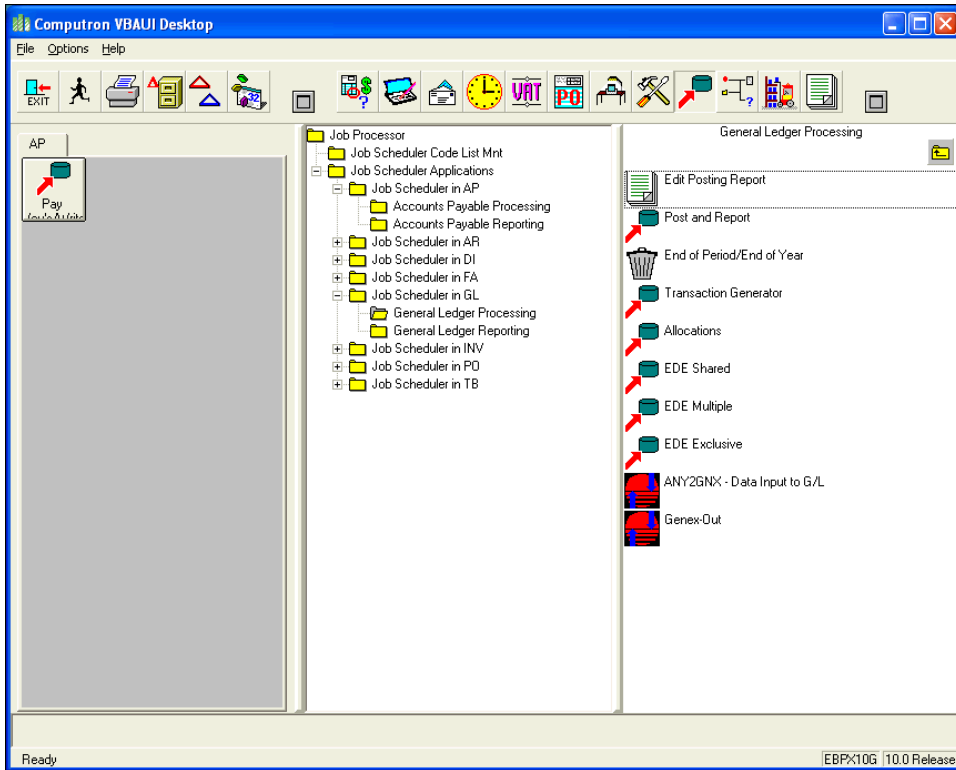
B – This value indicates that the Job Scheduler Maintenance Window (BS1) is to display. However, by pressing the Continue pushbutton on this window, the user can bypass this window and continue with the other windows in the function.

Blank – This value indicates that the Job Scheduler Maintenance window (BS1) is not to display: the user is to proceed directly to the other windows in the function, to select options and initiate processing as usual.

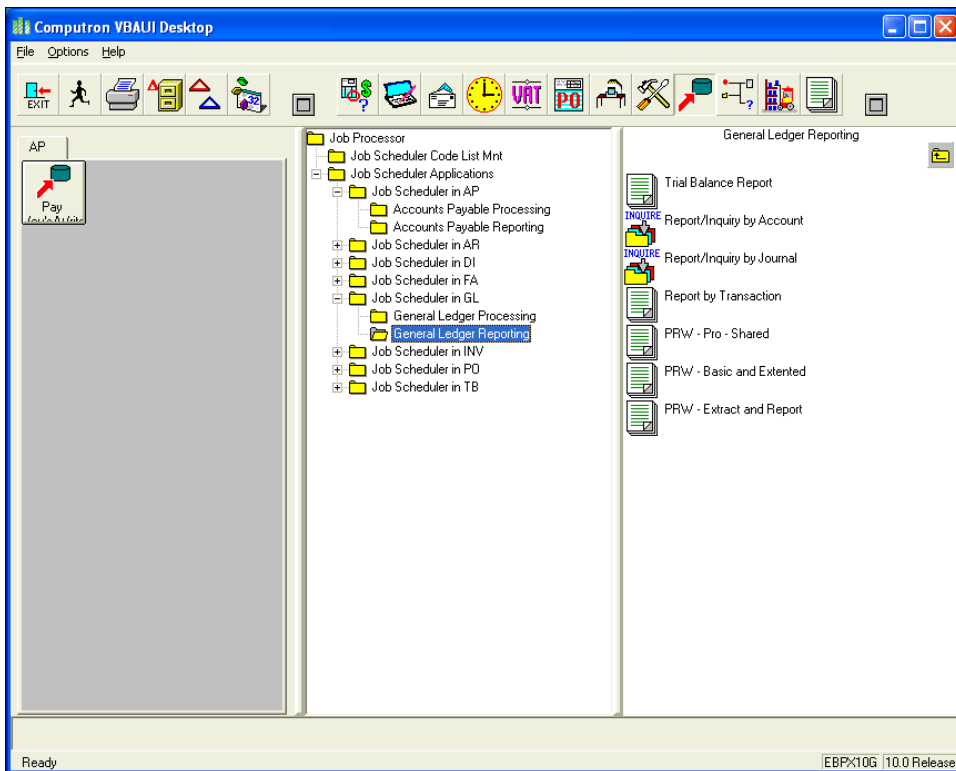
Job Processor Only Menus

This approach allows you to maintain access to existing Computron functions on an as needed basis, while providing access to establish parameters for these same functions under a specific Job Scheduler Only menu. When the system is initially installed, standard menus are released for each Computron application system. These menus include those functions that are most commonly processed via the Job Processing module, as illustrated here.

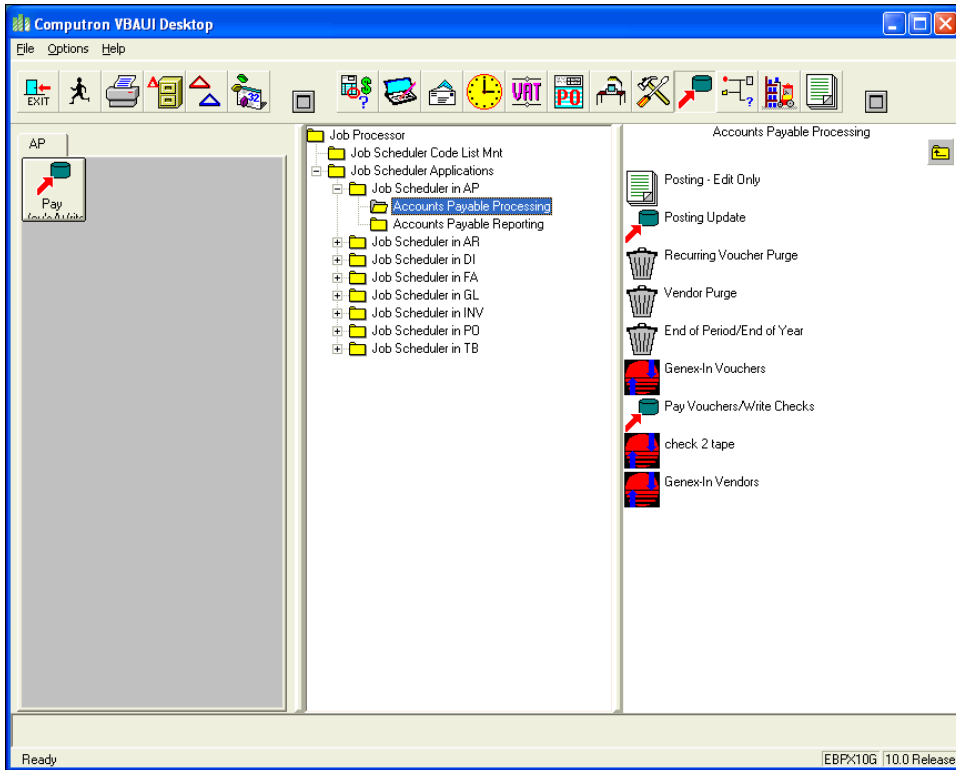
Note: These menus are released with the system, but can be modified or replaced completely at a client site. In addition, non- Computron functions can be placed on the menu and parameters established for processing using the same method as for Computron functions.



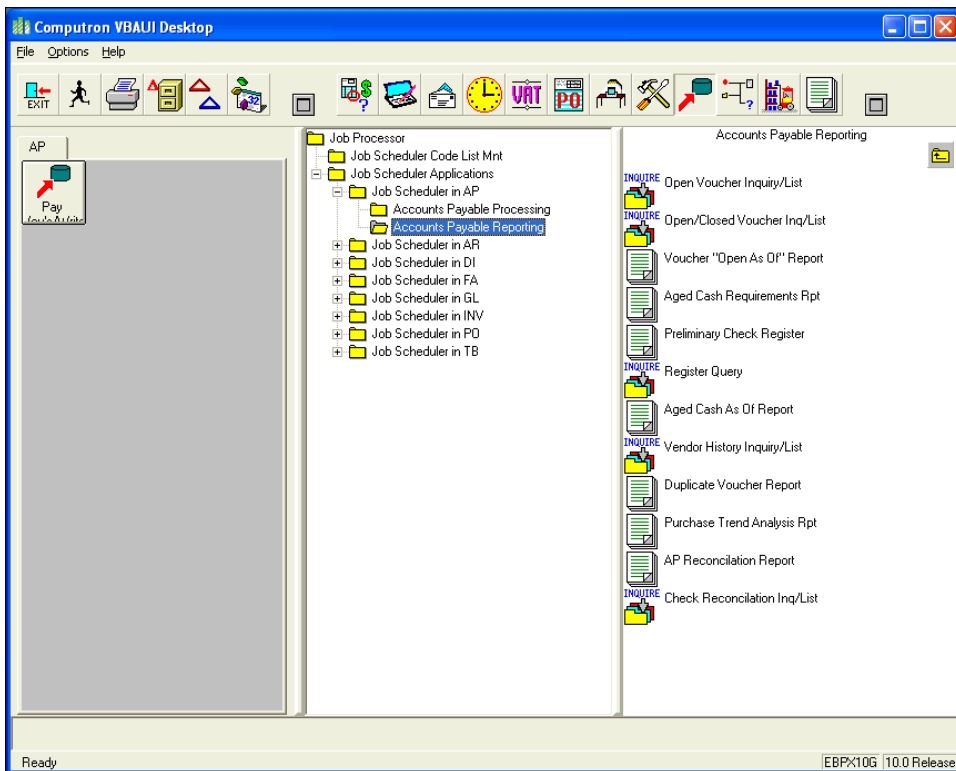
GL Processing Functions



GL Reporting Functions



AP Processing Functions



AP Reporting Functions

3.12 Return Codes

Introduction to Return Codes

The implementation of return codes within Computron's application software enables complex routines, which are dependent upon the successful completion of successive programs, to drive the Computron system. The following sections are intended to provide the user with an understanding of the processes and the value of return codes. Included in this documentation are descriptions of the sequence of events that occur when running Computron application software, the setting of return codes and the Job Processor and its Return Code table.

The Computron Application

All Computron application software is launched by Computron's `start` script, which invokes Computron server or application software and initializes the environment of the logged-in user. The `start` script resides in `/.../exec/_ctrone_` (where `/...` represents the file system in which the Computron application software resides). The `start` script may be initiated interactively by a user or this routine can be called from a script (e.g., when the user logs onto the Unix system). The syntax of the command is as follows:

```
start [-s screen_type] [-i image_type] [-d database_type]
      [-c custom_prefix] [-f menu_file] [-v(u) port_number (d)] [ -w ] [ -p ]
      [ -g ] [ -h ] [ -m ] [ -b ] [-r runtime_path]
      [-n[+] configuration_filename] [ -x ] [program_name]
```

☞ *Note:* For more details regarding the `start` script, refer to section 5.7, "Computron's Start Script," later in this guide.

Unless the `-m` option is used, the script causes the specified COBOL program to be run. If a program name is not specified, the default program, `MENUPROG`, is run. `MENUPROG`, in conjunction with the arguments issued accompanying `start`, begins the Computron application software and displays the initial menu, thus entering Computron's menu subsystem. In character cell mode, application selections chosen from the menu system, invoke at least one Unix process. In AUI mode, however, an application process is created and then re-used.

This process can be initiated by the `CSLINK` routine or the `Computron` submaroon utility. The `CSLINK` routine creates new Unix processes as required. This happens, for example, when a front-end process creates a back-end process to run a posting or report. The `submaroon` utility enhances the efficiency of applications that normally create more than one process. `Submaroon` does not terminate a process, thus enabling the same process to be subsequently reused as opposed to being re-created. Computron employs the `submaroon` utility when the selected application option does not invoke a script; otherwise, the `CSLINK` option is used. It is possible, however, to run Computron application software without the `submaroon` utility since it can be disabled by setting the environment variable, as follows:

```
CTRON_DISABLE_SUBMAR=1.
```

When the application function utilizes the `CSLINK` routine (such as a front-end application function initiating a back-end function), several processes may be generated. Often, `CSLINK` invokes a script that runs several programs. Each program within the script is invoked by the routine, `WRUN`. Each time

WRUN is executed, a new process is created. The first program encountered creates one process. The executable is run in that process until it completes; then, its process terminates and control is passed back to the script. The script may execute WRUN again, thus creating a new process in which the executable is run. Each executable is run after the previous one completes and each executable is initiated by WRUN.

When a program initiated via WRUN is terminated by an operator using the keystrokes Ctrl+c, the script terminates as well; therefore, no other functions within the script are initiated. If a program initiated via WRUN from a script terminates with a disaster or is killed by operator intervention (kill -1), however, the script continues processing with the following step. Therefore, the script should execute the next step based on a specific return code from the previous step or the script should test and exit if it does not find an expected return code (i.e., it finds a return code value of 888, indicating the previous program was killed by operator, or 889, indicating the previous program terminated with Disaster).

☞ *Note:* The selected application function may employ CSLINK to call a program or a script. The above logic applies within a script that is initiated via CSLINK. This case may be true when a front-end program connects to a back-end script.

How Return Codes Are Set

Submaroon

When the selected menu application function engages the submaroon utility, the return code feature is not used. However, if a link to a program is attempted from the menu and it fails because the program does not exist, the submaroon utility fails to be initiated and the return code is set to 997. This return code value causes a message to be presented on the menu window. If a program within submaroon is canceled by an operator using the keystrokes Ctrl+c or terminated by Disaster, the return code is set accordingly (888 or 889) and submaroon exits.

CSLINK and WRUN

Whenever the CSLINK routine or the WRUN function is invoked, the following return code settings occur:

- ◆ The return code is assigned the initial value, 888.
- ◆ The successful completion of the initiated program or script sets a new return code with an appropriate value. (This value is application specific.)
- ◆ In the event of abnormal termination of any process within the selected function, (e.g., user intervention such as the kill -1 command) the return code value remains 888 and an entry is written to the log file indicating the ending status of the application function.
- ◆ If the application detects an unexpected condition and the Computron function CSDISASTER is executed, a return code value equal to 889 is designated and information regarding the error that prompted CSDISASTER as well as database access details are written to the log file.

☞ *Note:* Interactive front-end and data entry/maintenance programs do not necessarily set the return codes to a standard value. Most interactive programs assign a value of zero to the return code.

The Return Code Mechanism

Computron applications set return codes via the `setretcode` routine and retrieve return codes via the `retcode` routine. Procedures can also set return codes using the `return_code` command and retrieve return codes by using `wretcode` command. All Computron provided scripts set the return code upon exit. Usually, the return code from the most meaningful program within the script is used to set the return code. The most meaningful program is determined by each application; in the example of the AP Register Report (figure 1), GENEXGEN sets the return code upon successful completion of the process. If an application terminates abnormally, due to an unexpected return code value (e.g., 888 or 889), then the return code is set to the value of the unacceptable return code. Successful return code values include 000, 102, 103 and 105.

`Wretcode` returns only the most recent return code provided by the Computron application software when issued without any arguments. However, the Computron `wretcode` command can retrieve any of the last 10 return code values.

The `wretcode` command can be used to ask by line number or program name (both are optional). If you type `wretcode -?`, it tells you that:

```
Usage: wretcode [-d nn] [program_name]
```

The `-d` option causes `wretcode` to retrieve the return code that is `nn` deep in the stack of return codes. The optional program name may be used to get the last return code from a specific program no matter where it appears in the return code stack. (`[program_name]` is case-sensitive). When used in combination with the `-d` option, the `nn` function retrieves the return code for a specific program. The `nn` function defaults to 0. The number 1 returns the next code in the stack and so forth.

The Job Processor

A stand-alone model of Computron's Job Processor (BSJOBPP2) can be proceduralized in a similar fashion to the Computron application software. (Also refer to Figures 3 and 4.) Beginning with the start routine, a script that invokes the Job Processor issues the start command with the `-m` and `-b` options. The `-m` option initializes the operator's Computron environment and the `-b` option initializes the environment for background processing.

Note: The start routine executes the `local.start` file for all users. Prior to beginning the application software, any commands contained in this file are performed for all Computron users. When issued with the `-m` option, the start routine executes the `local.init` file for all users. When run with this option, start initializes the user's Computron environment, but does not launch the application, display menu or execute the `local.start` file. Use the `local.init` file in place of the `local.start` file when issuing the start command with the `-m` option.

For more information about the start routine, and the `local.start` and `local.init` files, refer to Section 5.7, "Computron's Start Script," later in this guide.

The WRUN program is then called to initiate the Job Processor (JP) program. Remember that WRUN initially sets the return code to 888. The Job Processor initiates a back-end program or script using CSLINK. The return code settings outlined in the section describing how return codes are set in CSLINK apply within the Job Processor.

As each program completes (provided it runs in the same ground–foreground or background—as the Job Processor), the return codes set by the script are written to the Job Processor return code table by the Job Processor. Note that return codes indicating failure of a process (i.e., 888 and 889) appear in the Job Processor return code table as CNCL. A table describing each possible return code in the Job Processor return code table follows.

Standard Return Codes Set by Computron Applications

The table below shows the possible return codes as they are assigned by the application. Each application can vary the first digit of a return code; consequently, only the two right-most digits of a return code are guaranteed to be standard. Since only the last two right-most digits remain constant, only those digits should be tested in scripts, etc.

If return codes are checked in the job stream, masking may be used to specify the return code range so that only the last two characters are tested. For example, the return code range: From `**03` To `0003`, replaces the first two characters (those denoted as `*`) in the From field with the value of the same characters in the To field. The masked characters (as specified in the From field) are similarly replaced in the return code field from the Job Processor Return Code file before the compare is performed; therefore, a return code value of `0103` in the Job Processor Return Code file becomes `0003`.

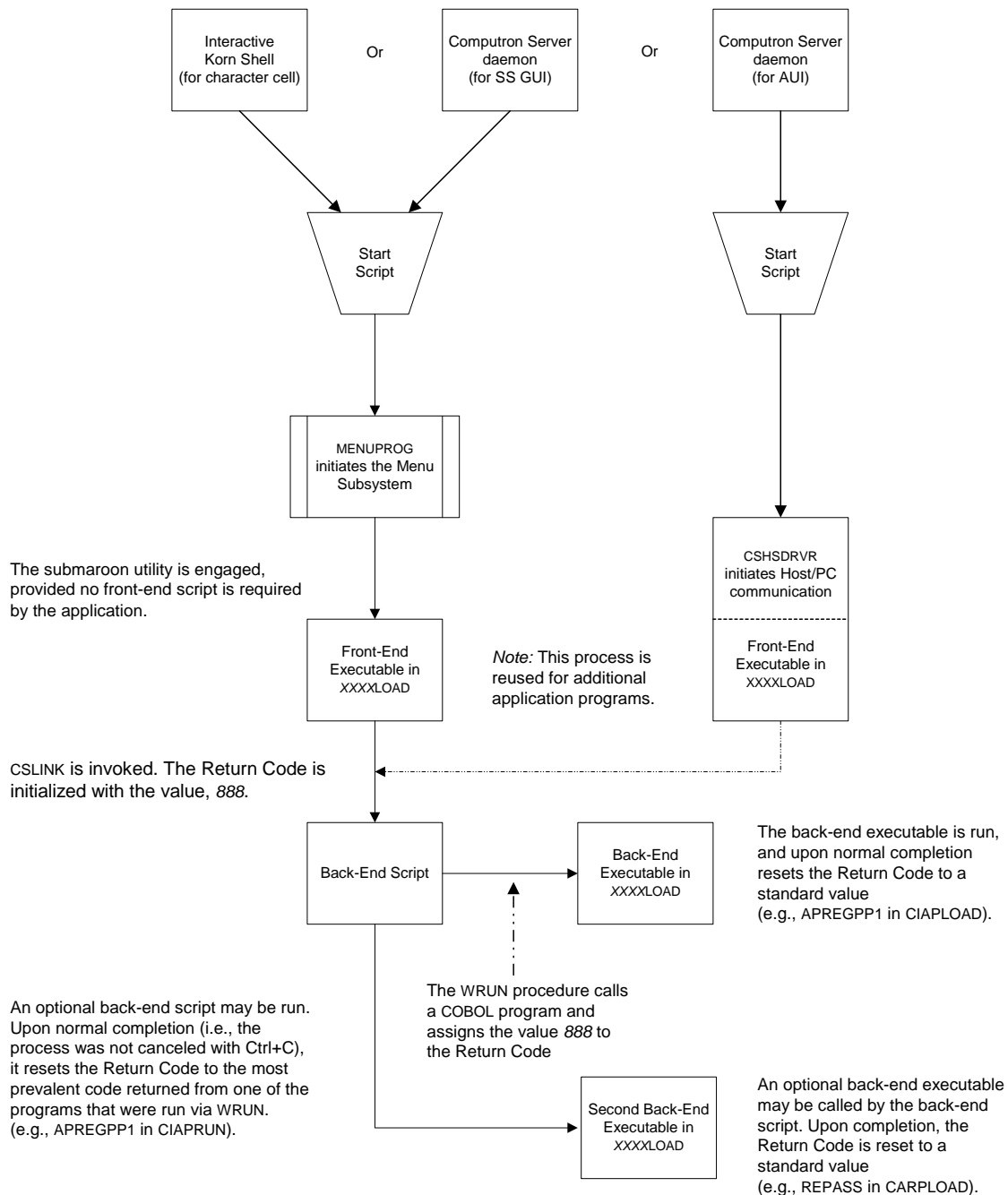
Return Code	Description
888	Process canceled by operator.
889	Process canceled by CSDISASTER.
997	Process canceled due to submaroon failure.
999	Link failed.
2101	Previous job was not run. Job Processor record not found. Job skipped (BSP only).
8101	Previous job was canceled in the Job Processor. Job skipped (JP only).
1101	Return Code not in range checked. Job skipped (JP only).
000	OK.
105	OK, but no records found.
102	OK, but warnings found.
101	Errors encountered.
111	Errors encountered evaluating the selection criteria or other values.
123	Print Pass record not found.
124	Operator has no access to program or report.
125	Program or report was not available (locked by another user).

Return Code	Description
126	Program or report was not found.
131	Program was interrupted by user.

Illustrations

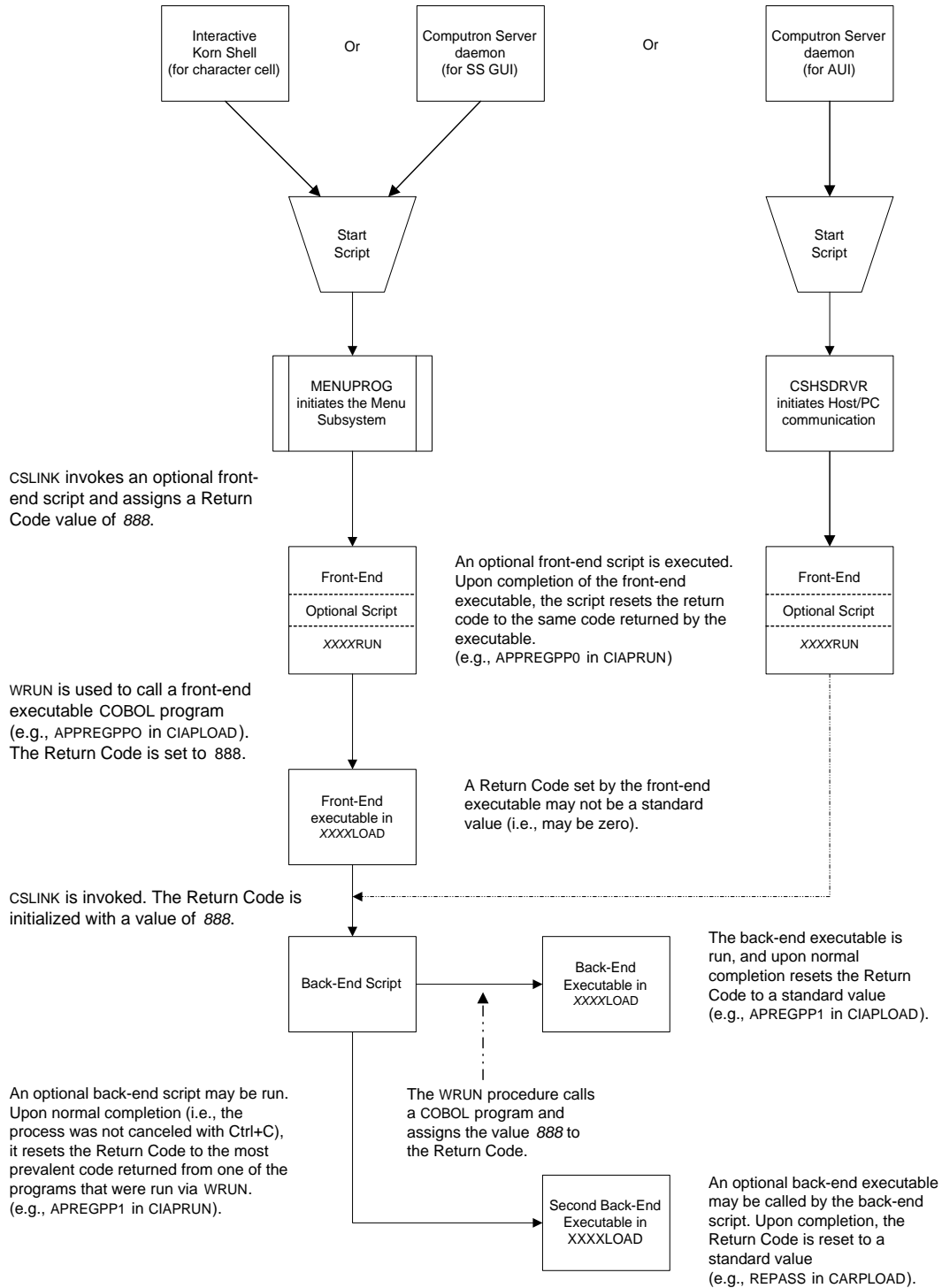
The flow chart below illustrates the flow of processes generated interactively, using the submaroon utility.

Figure 1: Application Process Flow Using Submaroon



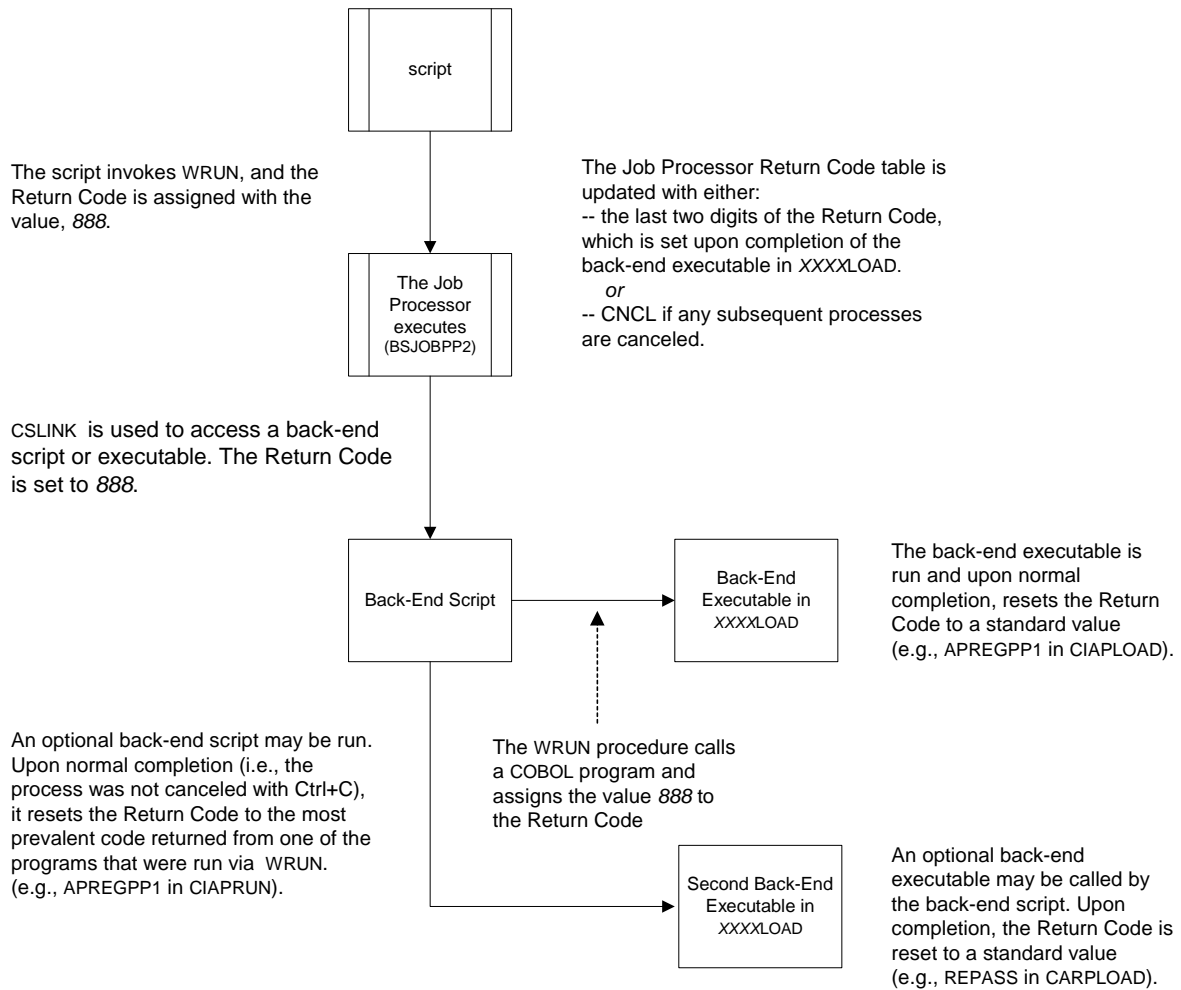
The figure below represents the flow of processes generated interactively, when the MENUPROG program invokes a script using CSLINK.

Figure 2: Application Process Flow without Submaroon or with Front-End Script



The following diagram illustrates the flow of processes generated by the Job Processor when initiated via a script.

Figure 3: Sample Standalone Job Processor Process Flow



The following script serves as an example of a script that can be used to initiate the Job Processor from outside the Computron software environment.

Figure 4: Sample Script for Invoking the Job Processor

```
# If this procedure (UNIX shell script) runs outside of Computron's software
# (i.e. standalone), uncomment the next 3 non-blank lines and set
# CTRON_EXEC to the correct directory
#
CTRON_EXEC=/ctronprod/exec
. ${CTRON_EXEC}/_ctron_/start -b -m [other start options]...
create_worklib

csputparm ENTER CSHPSET OPTION=G

csputparm ENTER KS1 PFKEY=02 DATEDTC="19941101" PAUSDTC=" " PAUSEHH=" " \
  PAUSEMM=" " AMPM=" " INDGRP=" " INDID=" " CLASS=" " CLASS02=" " \
  CLASS03=" " CLASS04=" " CLASS05=" " CLASS06=" " CLASS07=" " \
  CLASS08=" " CLASS09=" " CLASS10=" " FREQCODE=" " FREQCO02=" " \
  FREQCO03=" " FREQCO04=" " FREQCO05=" " FREQCO06=" " FREQCO07=" " \
  FREQCO08=" " FREQCO09=" " FREQCO10=" " APPLICD=" " JOBCAT=" " \
  GROUND="D" RUNMODE="D" CALENDAR=" "

csputparm ENTER GTUSEROV PROGVOL=EXEC
wrun bsjobpp2
CTRON_PROG_RETURNCODE=`wretcode`
export CTRON_PROG_RETURNCODE
csputparm clear
```

Chapter 4

System Tools

4.1 CSDISPLY – Computron's Display Utility

Introduction to CSDISPLY

Computron's display utility, CSDISPLY, allows you to display and print data accessed by Computron financial applications. You can display Computron data stored in relational database tables, print files, consecutive files, and indexed files. All data associated with a selected RDBMap file is merged and logically sorted before it displays; consequently, you can easily review related data that is stored in multiple tables.

CSDISPLY facilitates data analysis and provides many data display options:

- ◆ Review data in either character or hexadecimal format or examine data in both formats simultaneously;
- ◆ Select particular data description list (DDL) fields associated with an RDBMap file to display specific data in a readable format;
- ◆ Specify sorting preferences based on selected DDLs or table indexes;
- ◆ Search the display data for a particular value;
- ◆ View packed or binary fields in unpacked format.

Generally, CSDISPLY is restricted to users with a high security level. To run this program, click the Run any Program button on the Desktop toolbar. Please contact Computron support for instructions on using this powerful data analysis feature.

4.2 vutil – An Indexed File Utility

Introduction to vutil – An Indexed File Utility

vutil is an ACUCOBOL utility program that allows you to view and manipulate indexed files in vision file format. It can be used to create empty data files, examine files, extract data records and rebuild corrupted indexes. The utility is also used to bring data from one hardware platform to another or to troubleshoot and correct any incidents of file corruption. Generally, vutil is restricted to users with a high security level.

The default version of vutil currently used is part of ACUCOBOL version 4.0. This version allows vision files to exceed the previous 2GB limit. When vision files are created using the 4.0 version of vutil, 2 files are created, the vision file with the file name specified in the fdl and a corresponding .vix file that has the same name as the vision file with an extension of .vix. This .vix file contains index information for the vision file and the vision file cannot be read if the .vix file is not there. Therefore, it is critical that when copying or moving vision files, the corresponding .vix file be copied or moved as well.

Enter vutil at the Unix command prompt to list all commands. The commands are described in detail in the remainder of this section.

```

Enter 'exit' to return to Computron application
supsun2:/export/home/pse> vutil
vutil -info [-kpx] [files]
      -check [-af] [files]
      -zero [files]
      -rebuild [-acdlt234] [-bfgk #] [+ce] [files]
      -gen [-234] [filelist directory]
      -size [files]
      -extract [-x] file
      -load [-(b|d|t)>lnrv] source-file dest-file
      -unload [-(b|t)>u] source-file dest-file
      -convert [-acd234] [-f #] [+c] [files]
      -tree file
      -version

Enter 'exit' to return to Computron application
supsun2:/export/home/pse> _

```

vutil – Indexed File Utility

Commands

The following conventions apply to the syntax examples shown in this section:

- ◆ [] – The information entered within brackets is optional.
- ◆ User defined information is written with underscores (_), e.g. file_names.

check

This option of vutil tests a file for internal consistency. Tests are performed for a nonzero user count and other quickly tested errors. When options are entered with the command, additional tests are completed.

- a The automatic option initiates a test of any files in the directory detected with a nonzero user count and other quickly tested errors. Each record of the files is read to check for broken files. Broken file results are printed.
- f This option is similar to the –a option, except all records are examined.

☞ *Note:* Upon completion of vutil -check, a status value is returned to the host operating system. Possible values are:

Status Value	Description
0	File passed all tests
1	Command not fully executed because the file was in use.
2	Non-zero user count detected.
3	File is corrupt
255	vutil fatal error or incorrect command line.

Syntax: vutil -check [-a] [file_names]

☞ *Note:* If a list of files is checked, the highest status value that applies is returned.

convert

This function converts RM/COBOL-85 or C-ISAM files into Vision files. Several options are available to specify how the file is converted.

- # This option indicates which vision file format to use for the newly converted file. # indicates the version of ACUCOBOL to use: –2 indicates version 2 and –3 indicates version 3. A command without a version option converts the file to version 4.
- a This option suppresses an interactive message. Normally, vutil requires confirmation of the vutil convert command before processing occurs.
- c This option results in uncompressed records.
- +c This option compresses the records of a newly converted file.
- d This option places the new files in a new directory. This option is always followed by the directory name of the destination for the new file and it is the only option that does not destroy the original files.

Syntax: vutil -convert [-3] [-ac] [+c] [-d directory_name]
[file_names]

extract

This allows the user to print selected records. The system prints a synopsis of the file and prompts the user to indicate which key to read, a starting value, and total records to print.

Syntax: vutil -extract [file_name]

☞ *Note:* vutil does not extract records from encrypted files.

gen

This creates empty vision files. Attributes of the new file can be stored in a file that is called by the `vutil -gen` command or entered interactively by using the appropriate syntax of `vutil -gen`.

`-#` This option indicates which vision file format to use for the newly converted file. `#` indicates the version of ACUCOBOL to use. `-2` indicates version 2 and `-3` indicates version 3. A command without a version option generates the file in version 4.

☞ *Note:* As of Computron release 6.0, Acucobol Version 4.0 has become the default version of Acucobol. If no version number is specified, the version 4 file is created with the corresponding `.vix` file.

Syntax: `vutil -gen [-3]` (This is the Interactive version.)

Or:

`vutil -gen [-3] file_list_directory`

Attributes can be assigned to a newly created file. Computron uses the `.fdl` file to identify these attributes. They include:

- ◆ name of the file;
- ◆ blocking factor of the file;
- ◆ number of blocks to pre-allocate for the file;
- ◆ number of extension blocks to allocate (This number determines how many blocks to allocate when the space initially allocated is full.);
- ◆ compression factor, where 0 indicates no compression, 1 is the default compression and 2 – 100 represent percentages;
- ◆ enable encryption;
- ◆ maximum record size (This number cannot be greater than 32,767);
- ◆ minimum record size (If the minimum record size equals the maximum record size, the records are fixed length.);
- ◆ number of segments in the primary key (For Computron, this is always 1);
- ◆ segment size in bytes;
- ◆ segment offset from the start of the record in bytes;
- ◆ number of segments in each alternate key, where applicable;
- ◆ whether duplicate keys are allowed;
- ◆ segment size of the alternate keys, in bytes;
- ◆ segment offset from the start of the record, in bytes;
- ◆ name of a file containing a translation table (This table is used to determine the collating sequence for the keys to the file. Unless a collating sequence is specified, vutil uses the standard ACSII collating sequence.);
- ◆ comment of up to 30 bytes.

info

This displays general information about the file including:

- ◆ The comment line,
 - ◆ The Vision version number of the file,
 - ◆ The number of records contained in the file,
 - ◆ The number of deleted records in the file,
 - ◆ The size of the file,
 - ◆ The minimum and maximum record size within the file,
 - ◆ The number of keys,
 - ◆ The user count.
- x This option provides extended information about the file.
- ◆ The key size,
 - ◆ The number of segments,
 - ◆ The key offsets;
 - ◆ Whether duplicate keys are allowed;
 - ◆ The block size of the file,
 - ◆ The number of blocks per granule,
 - ◆ The tree height of the file (minimum, maximum, and average figures),
 - ◆ The number of nodes and
 - ◆ The number of deleted nodes;
 - ◆ The total node space and
 - ◆ The node space used.
- p If more than one file is specified, the –p option causes vutil to pause between each file and prompt you to press the Enter key.

Syntax: vutil -info [-xp] [file_names]

load

The load option creates an indexed file from a binary sequential file. The binary file acts as the source file. The destination file must be an existing vision file. The system uses the vision file to determine the record length and key information. The newly indexed records are added to the end of this existing file.

- l The destination file is normally locked while the vutil –load command is executed. The file remains unlocked if the –l option is specified, thus allowing access to the Vision file.
- v This option indicates that records in the source file are variable in length.
- n This option creates a new Vision file. In this case, data in the existing Vision file is eliminated before the newly indexed records are loaded from the binary file.

Syntax: vutil -load [-lnv] source_file destination_file

rebuild

This rebuilds or recreates an indexed file.

- b This option sets a new blocking factor for the file. The –b option must be followed by a number indicating the new factor, 1 or 2 disk sectors per block.
- g This option sets a new extension factor for the file. The –g option must be followed by a number indicating the new factor, 1 or 2 disk sectors per block.
- # # indicates the version of ACUCOBOL to use. –2 specifies version 2 and –3 indicates version 3. A command without a version option rebuilds the file to version 4.
- a This option suppresses an interactive message, automatically replacing the original file with the new one. Normally, vutil requires confirmation of the vutil -rebuild command before processing occurs.
- c This option results in uncompressed records.
- +c This option compresses the records of a newly converted file.
- +e This option invokes record encryption.
- f The option sets a compression factor for the file. The –f option must be followed by a number indicating the percent of compression.
- k This option rebuilds the file in key order. The –k option must be followed by the number of the key. (0 indicates the primary key and 1 indicates the first alternate.)
- d This option places the new files in a new directory. This option is always followed by the directory name of the destination for the new file and it is the only option that does not destroy the original files.
- s This option spools records to a file and rebuilds the file over itself. This option is often used when disk space is limited since the spool file can be located on a tape or disk drive. The spool file must be specified with the –s option.
- r This option allows the user to recover an interrupted rebuild. It is used in conjunction with the –s option. When used, vutil skips writing records to the spool file and begins rebuilding the file.
- m This option specifies the size of the spool media in bytes. –m is also used in conjunction with the –s option.

Syntax: vutil -rebuild [-b#] [-g#] [-3] [-ac] [+ce] [-f#]
 [-k key_number] [-d directory_name]
 [-s spool_file(-r)(-m spool_size)] [file_names]

size

This prints the total size of each file specified, the number of records in each file and the number and percentage of deleted records the file contains.

Syntax: vutil -size [file_names]

tree

This creates a file called v tree. This file contains a listing of the system's internal B-tree.

Syntax: vutil -tree [file_name]

unload

This command creates a binary sequential file from a vision file. The vision file (source file) and the binary file (destination file) must be specified.

`-v` This option indicates that records in the source file are variable in length.

Syntax: `vutil -unload [-v] source_file destination_file`

☞ *Note:* The unload command cannot be used with encrypted Vision files.

version

This command displays the version of the ACUCOBOL File Utility in use.

Syntax: `vutil -version`

zero

This resets the user count to zero for each file listed.

Syntax: `vutil -zero [file_names]`

4.3 Program Report List

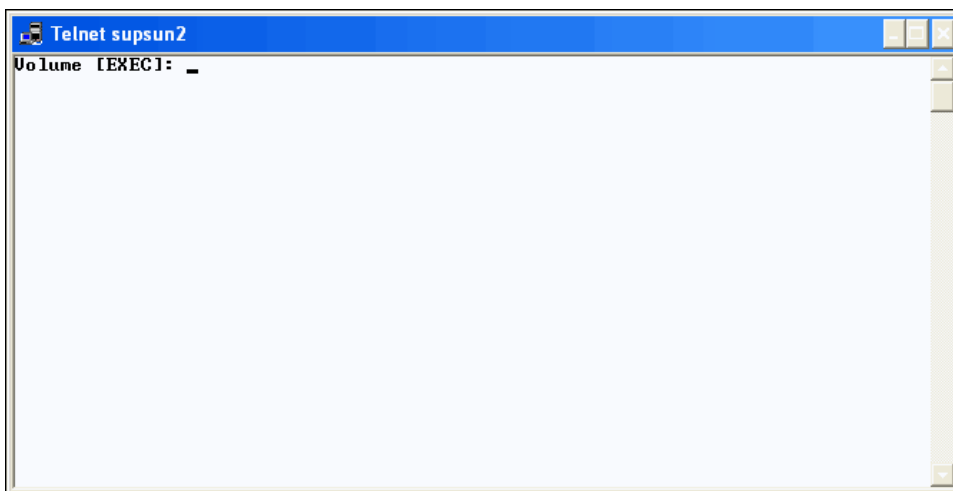
Introduction to Program Report List

This function generates a listing of all programs and subroutines in a specified volume/library. The listing is placed in your print queue. The following example demonstrates how to generate a program listing of the General Ledger load library.

☞ *Note:* This function is available via the character cell presentation or through ADMINTOOLS. For more information about ADMINTOOLS, refer to Section 5.1, “Computron Server Configuration Management – ADMINTOOLS.”

Program Report List Window

The following window displays when you select the Program Report List function from the System Utilities menu.



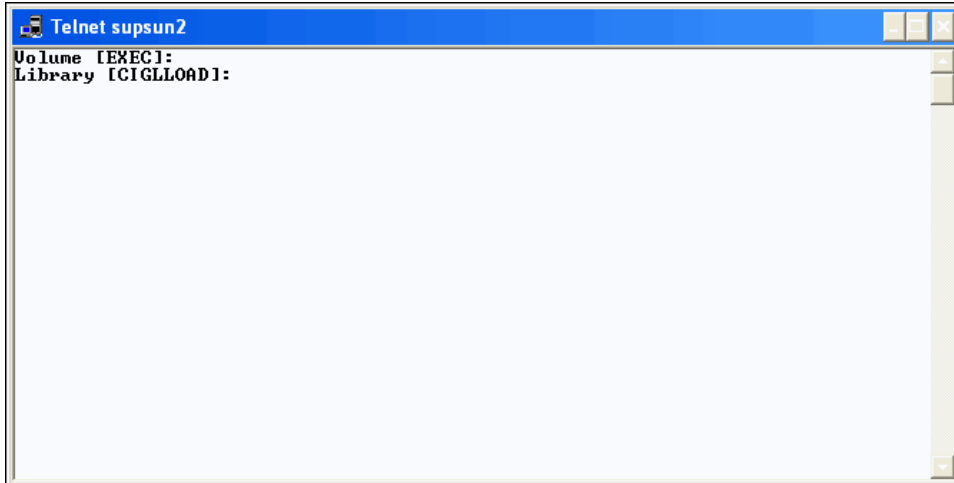
Program Report List Window

Volume

Enter the volume to interrogate.

Default: Current Volume.

Press enter to display the Library prompt:



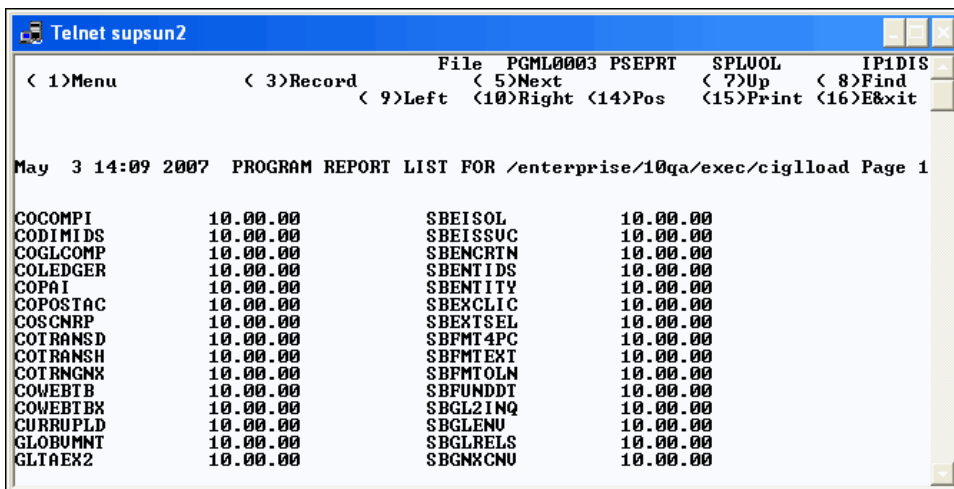
Program Report List Window

Library

Enter the library to interrogate.

Default: Current library.

Options for the Program Report List Window	
Function Key	Description
Enter	Generate a listing of all functions in the specified volume/library.



Online Inquiry — Program Report List Output

4.4 Display Computron Log Files

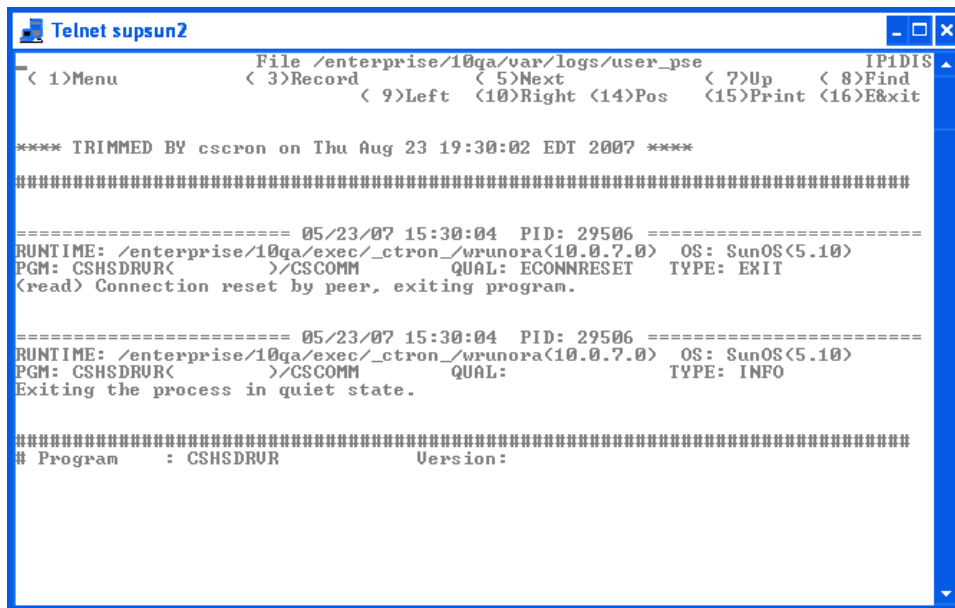
Introduction to Display Computron Log Files

This function displays the contents of your Log file. The Log file contains a running list of any error and/or informational messages that occur during a session. The Computron Log file is especially useful when analyzing system and/or program problems that produce error messages.

☞ *Note:* The Display Computron Log File function is currently only available under the character cell presentation.

Computron Log File Window

The following window displays when you access the Display Computron Log file function from the System Utilities menu:



```

Telnet supsun2
File /enterprise/10qa/var/logs/user_pse IPIDIS
< 1>Menu < 3>Record < 5>Next < 7>Up < 8>Find
< 9>Left <10>Right <14>Pos <15>Print <16>E&xit

**** TRIMMED BY cscron on Thu Aug 23 19:30:02 EDT 2007 ****

=====
===== 05/23/07 15:30:04 PID: 29506 =====
RUNTIME: /enterprise/10qa/exec/_ctron_/wrunora(10.0.7.0) OS: SunOS(5.10)
PGM: CSHSDRUR( >)/CSCOMM QUAL: ECONNRESET TYPE: EXIT
<read> Connection reset by peer, exiting program.

===== 05/23/07 15:30:04 PID: 29506 =====
RUNTIME: /enterprise/10qa/exec/_ctron_/wrunora(10.0.7.0) OS: SunOS(5.10)
PGM: CSHSDRUR( >)/CSCOMM QUAL: TYPE: INFO
Exiting the process in quiet state.

=====
# Program : CSHSDRUR Version:
    
```

Computron Log File Window

The Computron Log File Display window offers you a number of options that enable you to move through the Log file, and to locate specific data. In addition you can print a portion of, or the entire Log file using function key F15.

4.5 Clear Print Pass File

Introduction to the Clear Print Pass File Function

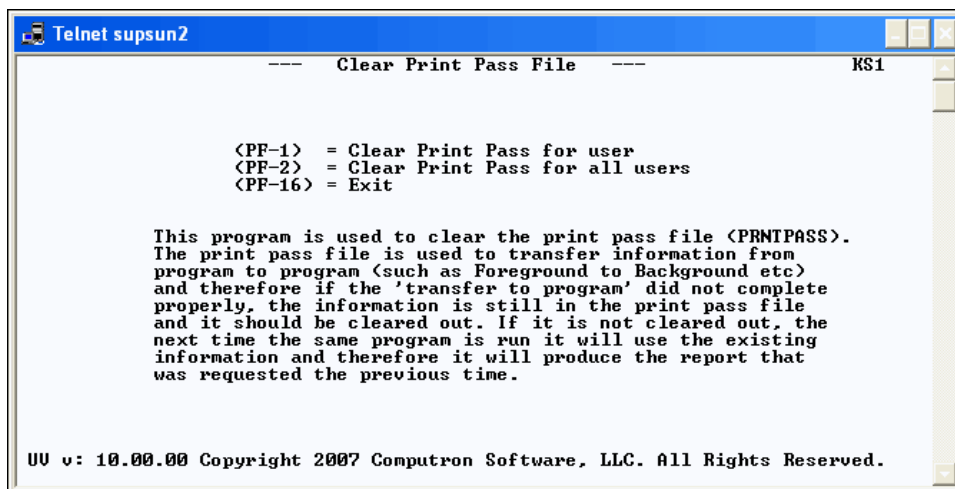
This function is used to clear Print Pass file (PRNTPAS4) records/entries created by various Computron functions. A Print Pass file is created for each database at the time of installation. This file contains one record for each function run in front-end/back-end mode; i.e., any function that allows you to enter selection criteria on a front-end window and subsequently calls a back-end program to perform the actual processing (usually posting and reports). In addition to identifying information in the Print Pass file record (e.g., program name and user ID), the record contains selection criteria and processing information. For the most part front-end/back-end functions are those functions that can run in either foreground or background.

Under normal processing circumstances, when any function uses the Print Pass file, the back-end program performs final clean up by deleting the corresponding Print Pass file record at the end. However, if some unexpected condition occurs prior to process completion (e.g., hardware failure, etc.), the Print Pass file may need to be cleared via this function.

Note: The Clear Print Pass File function is only available under the character cell presentation.

Clear Print Pass File Window (KS1)

The KS1 window displays when you select the Clear Print Pass File function from the System Utilities menu.



```

Telnet supsun2
--- Clear Print Pass File --- KS1

<PF-1> = Clear Print Pass for user
<PF-2> = Clear Print Pass for all users
<PF-16> = Exit

This program is used to clear the print pass file (PRNTPASS).
The print pass file is used to transfer information from
program to program (such as Foreground to Background etc)
and therefore if the 'transfer to program' did not complete
properly, the information is still in the print pass file
and it should be cleared out. If it is not cleared out, the
next time the same program is run it will use the existing
information and therefore it will produce the report that
was requested the previous time.

UU v: 10.00.00 Copyright 2007 Computron Software, LLC. All Rights Reserved.
    
```

Clear Print Pass File Window (KS1)

Clear Print Pass for User

This option is used to specify a single user for which the print pass file is to be cleared. The field following the text of the option is used only with the Clear Print Pass for user option PF-1. Enter the User ID in the field at the end of the option and select PF-1

Options for the Clear Print Pass File Window		
Function Key	Name	Description
PF-1	Clear Print Pass for user	Clears the print pass file records created for a specified user's ID.
PF-2	Clear Print Pass for all users	Clears the print pass file of all records, regardless of user ID.
PF-16	Exit	Returns to the Universal Utilities menu without making changes to the Print Pass file.

4.6 Clear Hold List Entries

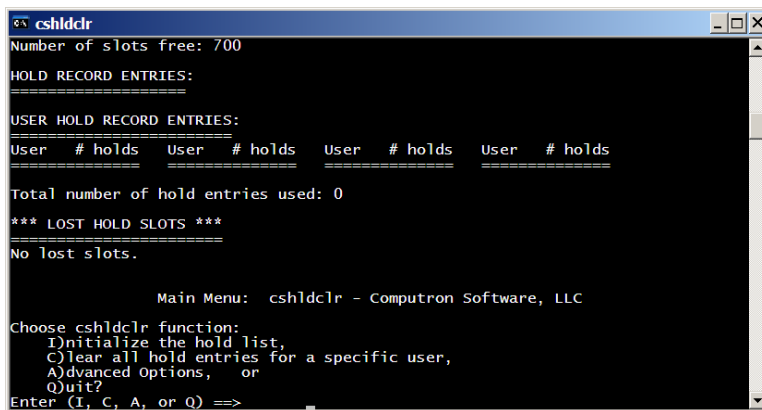
Introduction to the Clear Hold List Entries Function

The Clear Hold List Entries function (CSHLDCLR) removes held resources associated with Computron user IDs during runtime. These holds are automatically set by the system to protect the integrity of the database. For example, in cases where a user abnormally exits from specific functions (e.g., Pay Vouchers/Write Checks), the system cannot perform normal clean-up processing to clear the held resources; therefore, you can run the Clear Hold List Entries function to delete the hold list entry.

Note: This function is only available under the character cell presentation.

Clear Hold List Entries Window

The User/# holds window displays when you select the Clear Hold List Entries function from the System Utilities menu:



```

cshldclr
Number of slots free: 700
HOLD RECORD ENTRIES:
=====
USER HOLD RECORD ENTRIES:
=====
User # holds User # holds User # holds User # holds
=====
Total number of hold entries used: 0
*** LOST HOLD SLOTS ***
=====
No lost slots.

Main Menu: cshldclr - Computron Software, LLC

Choose cshldclr function:
I)nititalize the hold list,
C)lear all hold entries for a specific user,
A)dvanced Options, or
Q)uit?
Enter (I, C, A, or Q) ==>
    
```

Clear Hold List Entries

If held resources are pending, the system displays a list of user IDs together with the number of holds attached to each user ID. The options for clearing the holds are described below:

Choose cshldclr function:

- I)nititalize the hold list,**
- C)lear all hold entries for a specific user,**
- A)dvanced Options, or**
- Q)uit? (I, C, A, or Q)**

Enter (I, C, A, or Q) ==>

The answer to this question determines the hold clearing process. Valid entries are:

I – Clears all locks for all users. The following message displays:

```
"Other processes are attached to the hold list shared memory
segment. Do you wish to proceed (Y/N)?"
```

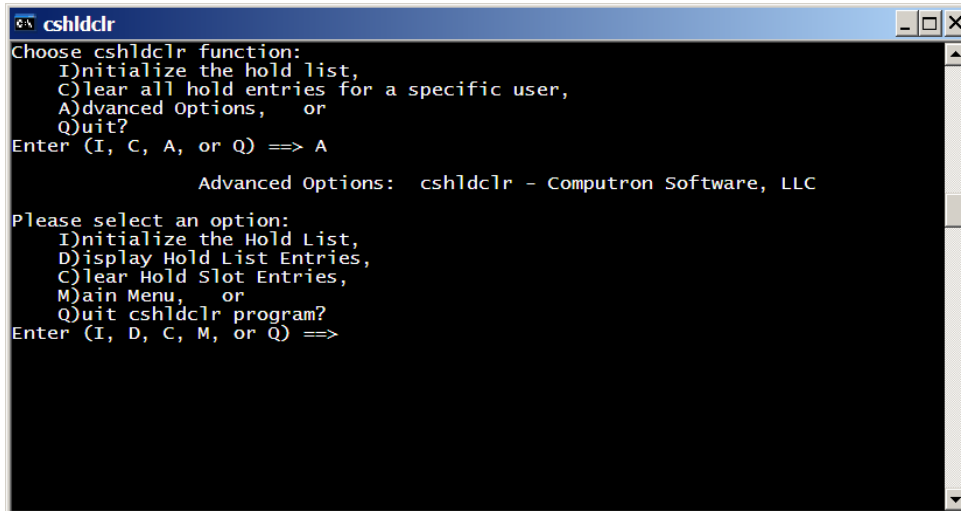
If all the holds are justified for clearing, enter Y to clear all holds for the listed IDs. Otherwise, enter N to return to the System Utilities menu.

C – Clears all holds for a specific user. The following message displays:

```
"Enter the user ID to clear:"
```

Enter the user ID that you want to clear. Upon completion; the system informs you whether the hold has been cleared for the user ID. Once the user ID is cleared, the system prompts you to return to the System Utilities menu.

A – Displays the advanced options menu:



```
cs cshldclr
Choose cshldclr function:
I)nititalize the hold list,
C)lear all hold entries for a specific user,
A)dvanced Options, or
Q)uit?
Enter (I, C, A, or Q) ==> A
          Advanced Options:  cshldclr - Computron Software, LLC

Please select an option:
I)nititalize the Hold List,
D)isplay Hold List Entries,
C)lear Hold Slot Entries,
M)ain Menu, or
Q)uit cshldclr program?
Enter (I, D, C, M, or Q) ==>
```

Clear Hold List Entries – Advanced Options

I – Clears all locks for all users. This is the same as the I option above.

D – Displays a list of all hold entries and optionally, displays details on hold list entries for a specific user or all users.

C – Clears all holds for a specific user. This is the same as the C option above.

M – Returns to Main Menu.

Q – (from either the Main Menu or Advanced Menu) Exits the Clear Hold List Entries function and returns to the System Utilities menu.

Tailoring the cshold Parameter File

The cshold parameters can be modified for a particular installation to accommodate your Computron environment. These parameters are established in the `/.../exec/_ctron/_ipc_parm` file (where `/...` represents the directory name where the Computron application software resides). The default configuration is set to use shared-memory hold lists with 700 hold slots allocated and a session hold limit of 250. These values can be changed by un-commenting the relevant command line and modifying the appropriate value. Each command line is preceded by a comment line that describes its function.

Sample IPC Configuration File

```
# The following line set the total number of hold entries to allocate.
#HOLD_SLOTS = 700
```



```

#
# The following line selects the number of hold entries per session to allocate.
#SESSION_HOLD_LIMIT = 250
#
# The next line selects the type of hold system (shared memory, disk_file, or none).
#HOLD_TYPE = shared memory
#
# The next line sets the path for Unix IPC functions – the file must exist.
#IPC_KEY_FILE = /ctron/exec/_ctron/_ipc_PARAM
#
# The following line is used only if HOLD_TYPE = disk_file.
#HOLD_FILE = /ctron/exec/_ctron/_hold_data

```

☞ *Note:* For debugging and testing purposes the environment variable `CTRON_IPC_KEYFILE` can be used to indicate a different `CSHOLD` parameter file than the default version.

☞ *Note:* There are other entries present in the `_ipc_param` file which are not related to hold processing. They are not shown here.

Viewing Active Process IDs

The `cshlddmp` utility also allows you to view (or dump) a listing of all the Computron process IDs that are currently active, including the user ID attached to each process.

To access `cshlddmp`, enter the following command line from a Unix prompt:

```
/. . . . /exec/csuvrun/cshlddmp
```

The `cshldini` Command

The `cshldini` utility is used to reinitialize the hold list entries on a regular basis. It does not require any user inputs and is generally run from Computron’s cron job – `CSCRON`. The `cshldini` utility prints informational messages on the standard output, so that unless messages are redirected to the `/dev/null` device, `CSCRON` related mail messages are generated on each invocation.

The following message should be ignored if `cshldini` is run by `CSCRON`, as it simply indicates that there is no hold list to clean up.

```

Hold listshared memory segment not found (key = ...)!
Shmget: No such file or directory.

```

☞ *Note:* For more information about `CSCRON`, refer to section 5.8, “Computron’s Cron Job – `CSCRON`,” later in this guide.

Chapter 5

Computron Server Administration for Unix

5.1 Computron Server Configuration Management - ADMTOOLS

Introduction

In the Computron Unix implementation, a tool is available for monitoring and collecting information about the Computron server environment via a Web Browser. The service runs on the Unix application and can be utilized by any client PC with access to a Web Browser and the application server. This tool offers an alternative to many of the manual methods that were required to analyze the application server in areas such as:

- ◆ Active application and csserve processes,
- ◆ Accessing the various log files,
- ◆ Checking different configurations,
 - ☞ *Note:* In order to view the configuration files, the user running ADMTOOLS must have a valid login ID and password to the server and is prompted for these when attempting to access the Configuration page.
- ◆ Collecting version information. This is only supported in Microsoft Internet Explorer Browser.

Starting the Service

Prerequisites

In order to start the ADMTOOLS service, there are certain requirements:

- ◆ The user starting the service must have access to the Unix shell command line.
- ◆ The user starting the service must have root access.

Starting the ADMTOOLS service

1. Login to the Unix server and get to the Unix shell command line
2. cd to the directory `$CTRON_EXEC/_ctron_/admttools`
3. From within the admttools directory, run the `set_perms.sh` script by typing the following:

```
./set_perms.sh
```
4. Exit from root
5. Type in the command "pwd." If not in the directory `$CTRON_EXEC/_ctron_/admttools`, cd to the directory `$CTRON_EXEC/_ctron_/admttools`
6. From within the admttools directory, run the `startadmttools` script by typing the following:

```
./startadmttools 1855
```

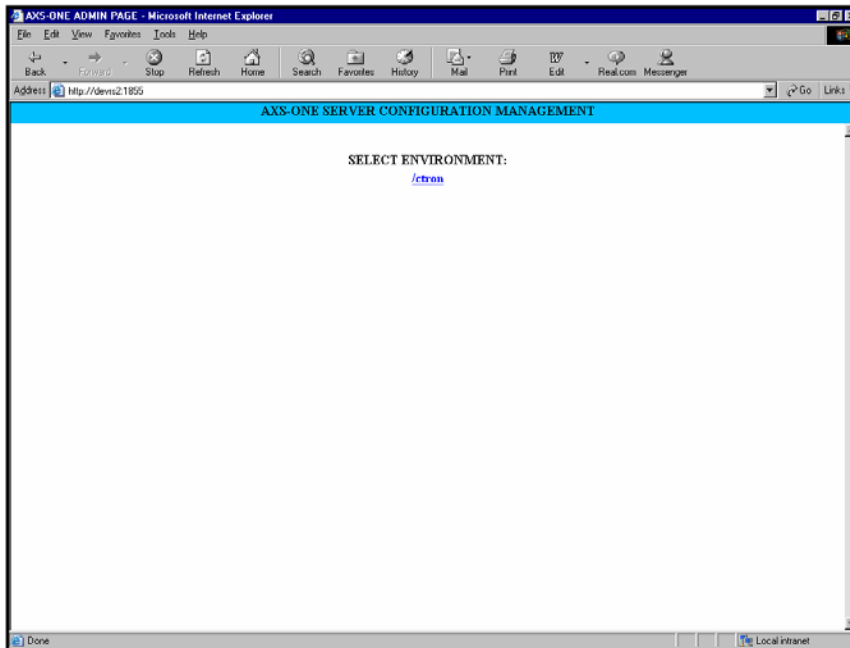
☞ *Note:* In the command above, 1855 represents the port number for the service and represents the convention for the admtools service.

Running the ADMTOOLS Utility from the Web Browser

1. From any PC that has Internet Explorer installed and has access to the application server, launch Internet Explorer.
2. In the Web Browser Address line, enter the following address:

`http://{server_name or IP Address}:1855`

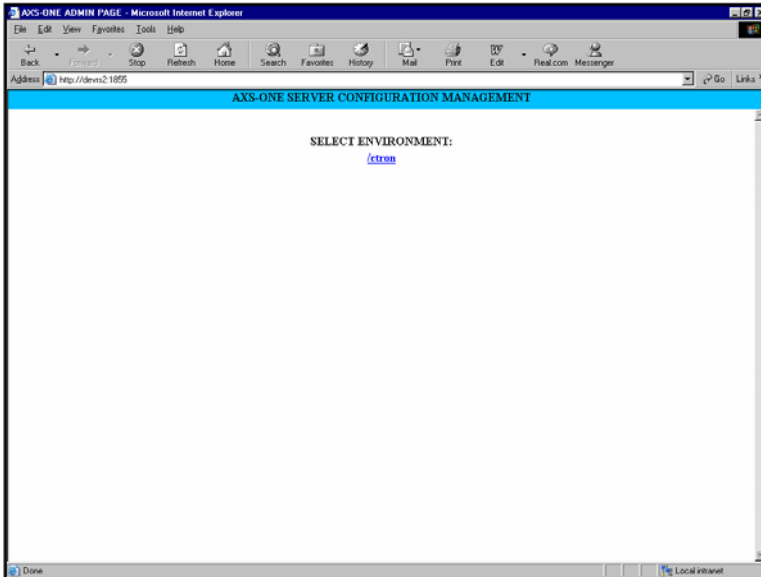
This loads the Computron Server Configuration Management Page.



Select Environment page

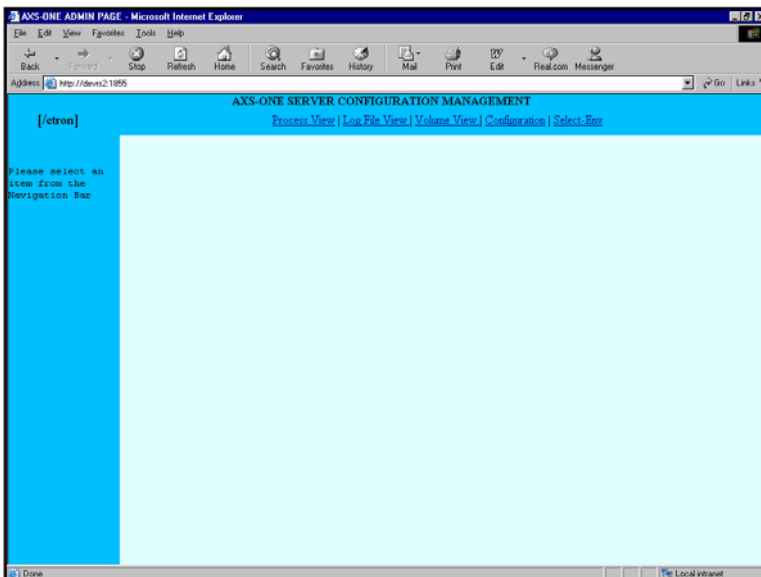
Using ADMTOOLS (Computron Server Configuration Management)

For each Computron software environment on a server, there is an option representing that environment on the Select-Environment page.



Select Environment page

Within an environment, there are a variety of options.



Computron Server Configuration Management Menu

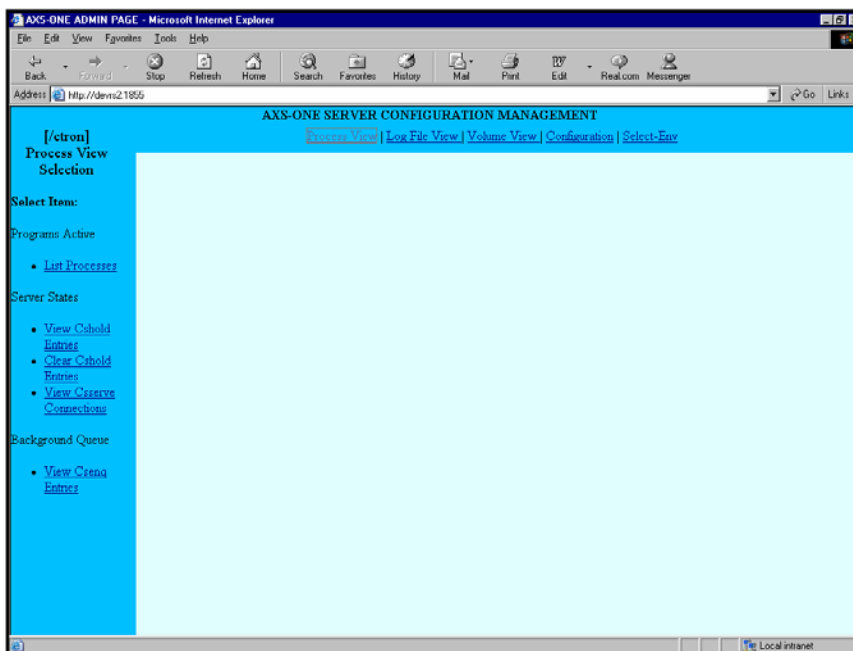
On this page, you have the following options:

- ◆ Process View – This option is the views of various processes/processing entries.
- ◆ Log File View – This option is the views of three types of logs: the system log, user logs and the cseq log.
- ◆ Volume View – This option is a view of the LGMAP.
- ◆ Configuration – This option accesses various setup configurations. To access these views, you must have a valid Computron logon ID. Once logged on, you can view or change the LPMAP, PRMAP, FORMS, LGMAP, send a printer test page, get version information for the various applications/products, csuvlpc information, runtime information, other utilities information, patch information and .config file information.

Each of these views is discussed in more detail in this section.

Process View

Within this option, there are 3 classifications and information options are presented within the classifications.



Process View Classifications and Options

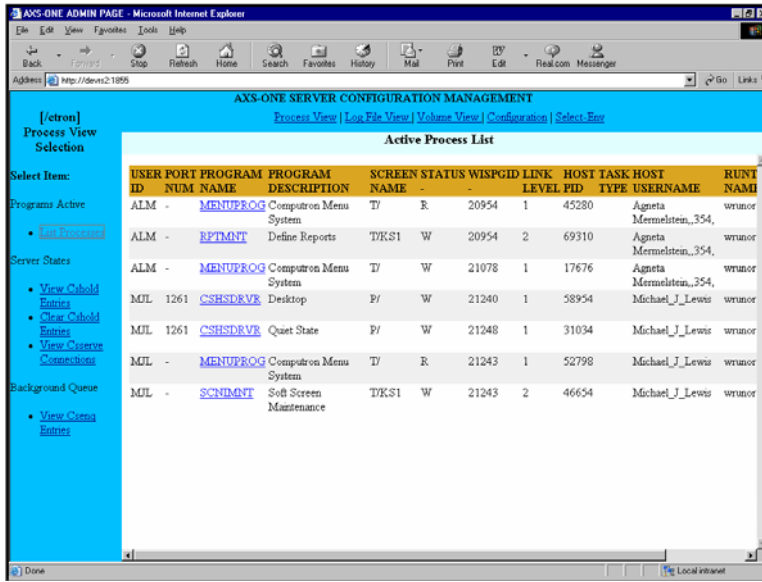
Programs Active

List Processes

This displays information about the processes associated with each active application program, listed by user. This is similar to running `ps -ef` from the command line, but displays

only processes associated with an application program. Some of the helpful values associated with a user's process that are shown on this page are:

- ◆ Program Name
- ◆ WISPGID
- ◆ Unix Process ID
- ◆ Runtime Name and Version



The screenshot shows a web browser window displaying the 'AXS-ONE SERVER CONFIGURATION MANAGEMENT' interface. The main content area is titled 'Active Process List' and contains a table with the following data:

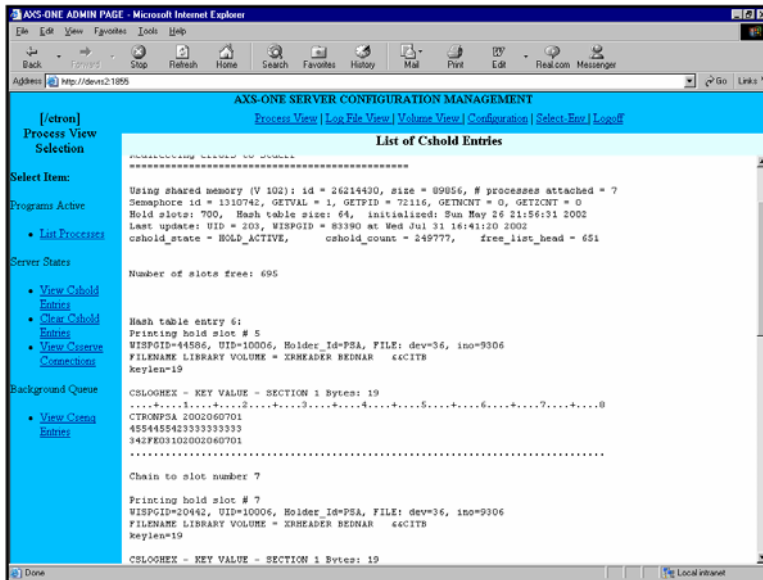
USER ID	PORT NUM	PROGRAM NAME	PROGRAM DESCRIPTION	SCREEN NAME	STATUS	WISPGID	LINK LEVEL	HOST PID	TASK TYPE	HOST USERNAME	RUNTIME NAME
ALM	-	MENUPROG	Computron Menu System	T/	R	20954	1	45280		Agneta Mermelstein_354,	wrunor
ALM	-	RPTMNT	Define Reports	TKS1	W	20954	2	69310		Agneta Mermelstein_354,	wrunor
ALM	-	MENUPROG	Computron Menu System	T/	W	21078	1	17676		Agneta Mermelstein_354,	wrunor
MJL	1261	CSHSDRVK	Desktop	P/	W	21240	1	58954		Michael_J_Lewis	wrunor
MJL	1261	CSHSDRVK	Quiet State	P/	W	21248	1	31034		Michael_J_Lewis	wrunor
MJL	-	MENUPROG	Computron Menu System	T/	R	21243	1	52798		Michael_J_Lewis	wrunor
MJL	-	SCNMNT	Soft Screen Maintenance	TKS1	W	21243	2	46654		Michael_J_Lewis	wrunor

Programs Active – List Processes

Server States

View Cshold Entries

This displays holdlists for this environment. This is similar to running `cshldmp` from the shell.



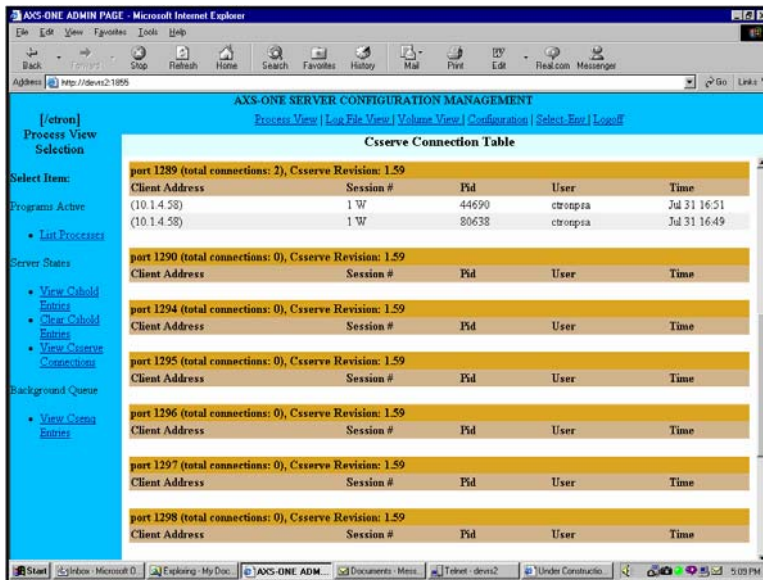
Server States – View Cshold Entries

Clear Cshold Entries

This clears holdlists for this environment. This is similar to running `cshldclr` from the shell.

View Cserve Connections

This displays the `cserve` processes for that environment. This is similar to running `ps -ef | grep cserve` from the command line.

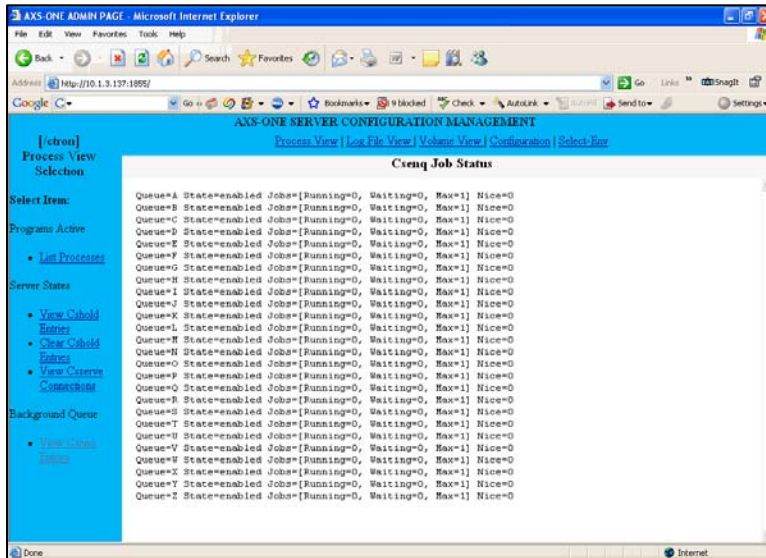


Server States – View Cserve Connections

Background Queue

View Csenq Entries

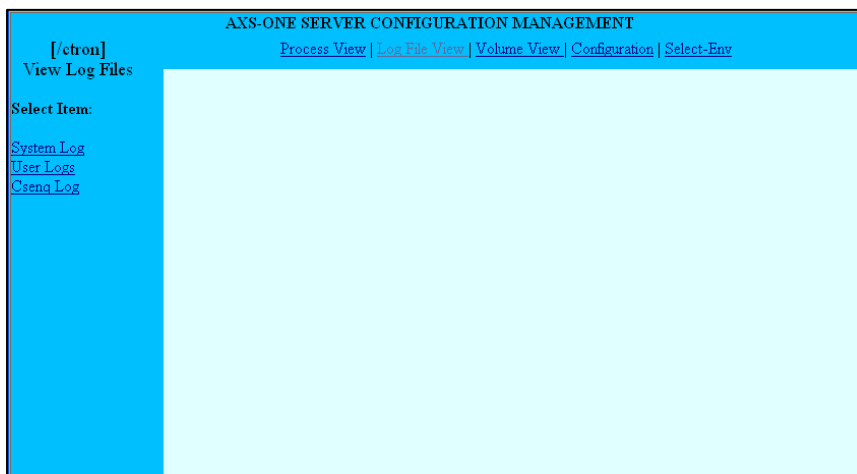
Displays background jobs when using the csenq implementation for background processing. This is similar to running `csenq -l` from the shell.



Background Queue – View Csenq Entries

Log File View

Within this selection, there are 3 logs that are available for viewing. The file name and full path of the file being displayed are also listed.



Log File View

System Log

This displays the existing system log file.

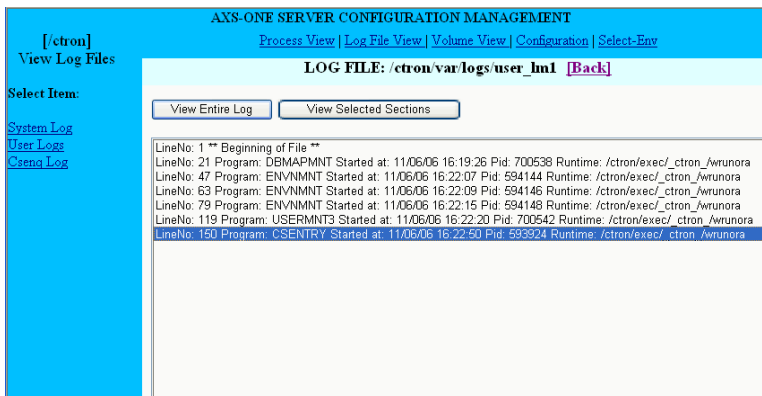
User Logs

This displays the list of existing user log files that can be selected and viewed. To access an individual user's log file, click the appropriate user_logon ID file name.

LOG FILE NAME	UNIX USER	UNIX GROUP	FILE SIZE	LAST MODIFIED
user_000	ctronsys	ctron	138	Jul 24 2001
user_amf	ctronsys	ctron	252121	Jul 25 17:38
user_bt1	ctronsys	ctron	26006	Aug 2 10:49
user_cs1	ctronsys	ctron	16383	Jul 30 17:47
user_ctr	ctronsys	ctron	2081	Aug 2 11:59
user_ctr.old	ctronsys	ctron	159302	Mar 21 11:39
user_drc	ctronsys	ctron	37543	Jun 19 16:49
user_edv	ctronsys	ctron	1654	Jun 4 08:15
user_eh	ctronsys	ctron	11181	Mar 27 13:15
user_fdl	ctronsys	ctron	534677	Jul 29 13:30
user_ke4	ctronsys	ctron	96793	Jul 19 08:45
user_lal	ctronsys	ctron	35121	Mar 25 16:55
user_lm1	ctronsys	ctron	94539	Aug 1 14:23
user_rlm	ctronsys	ctron	13374	Jun 17 13:52
user_pjc	ctronsys	ctron	28650	Jul 25 17:24
user_pse	ctronsys	ctron	213631	Jul 31 15:51
user_pt1	ctronsys	ctron	52220	Jul 29 17:26
user_rgh	ctronsys	ctron	252222	Aug 2 12:12
user_seh	ctronsys	ctron	28221	Jun 17 17:07

Log File View – User Logs

The individual user's log file can be divided into sections. When divided, you can select a section and view that section only. Otherwise, you can view the entire log file.



The screenshot shows the 'Log File View' interface for the file `/ctron/var/logs/user_lm1`. It includes a navigation menu on the left with options like 'System Log', 'User Logs', and 'Csenq Log'. The main area displays the log content with buttons for 'View Entire Log' and 'View Selected Sections'. The log content shows various system events and program starts.

Log File View – User Log Sections List

Csenq Log

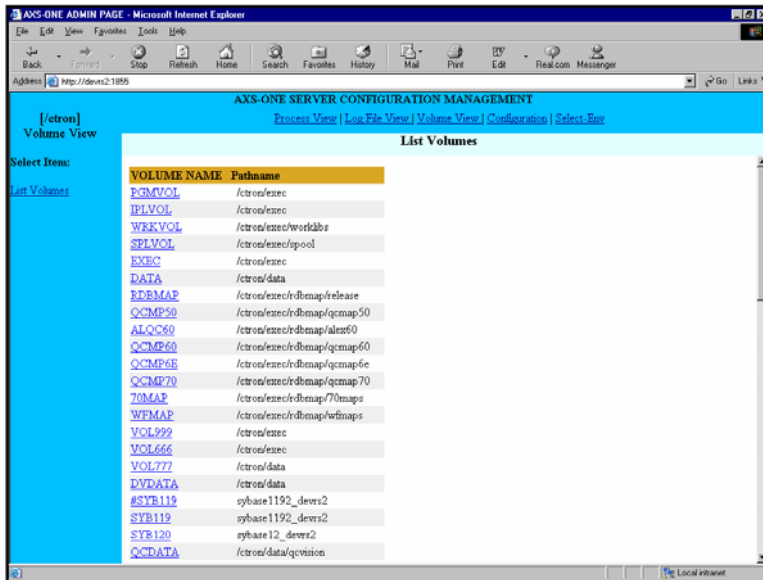
This displays the csenq log.

Volume View

Within this selection, there is one option available.

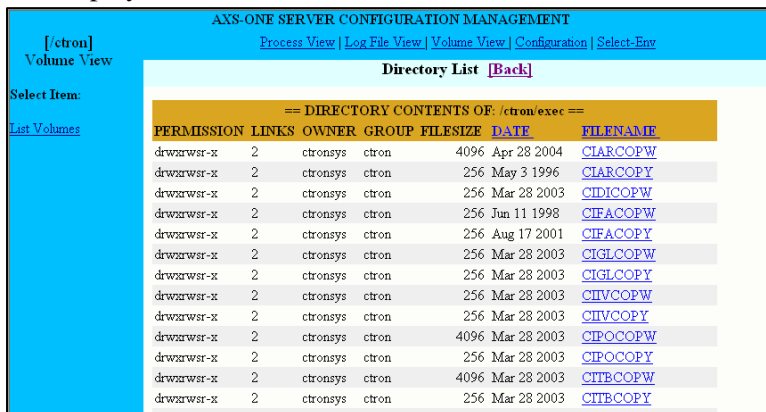
List Volumes

This presents the LGMAP entries and their associated full path. Click on a directory.



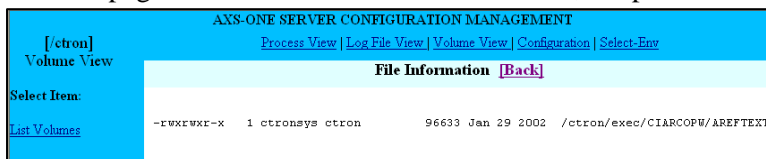
Volume View – List Volumes

Click on the Volume Name link and the contents of the directory represented by the path are displayed.



Volume View – Directory List

Click on the FILENAME link and the contents of a directory are displayed on a Directory List page. If the FILENAME link is to a file, the permissions for that file are displayed.

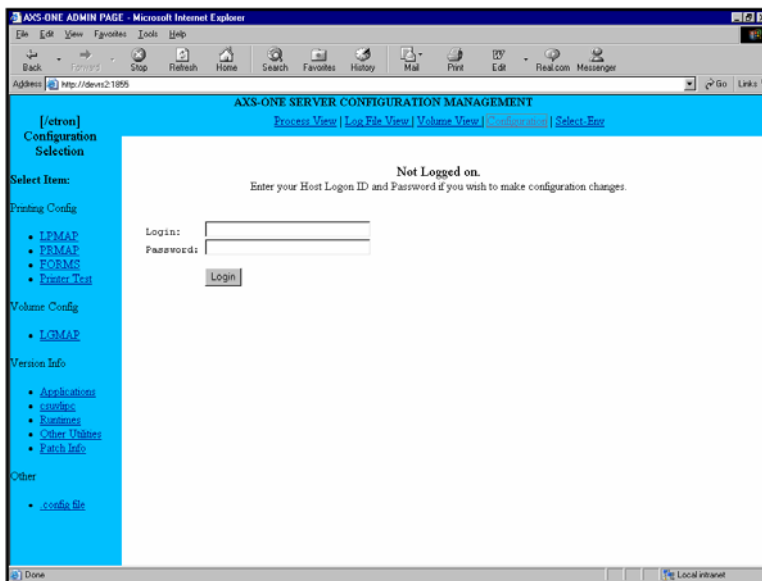


Volume View – File Information

Configuration

In order to access the various options under Configuration, it is necessary to enter a valid login ID and password. You must also have a csserve port running and valid permissions for this user. A login ID and password page is presented. Enter this information and press Enter. The available options are:

- ◆ Printing Config
 - LPMAP,
 - PRMAP,
 - FORMS,
 - Printer test page,
- ◆ Volume Config
 - LGMAP,
- ◆ Version Info
 - Applications,
 - csuvlpc,
 - Runtime,
 - Other utilities,
 - Patch information
- ◆ Other
 - .config file information

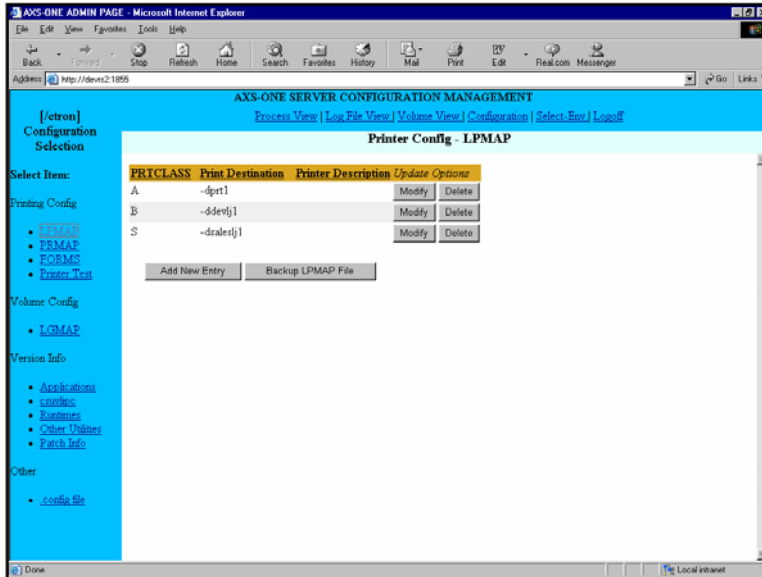


Configuration – Logon Screen

Printing Config

LPMAP

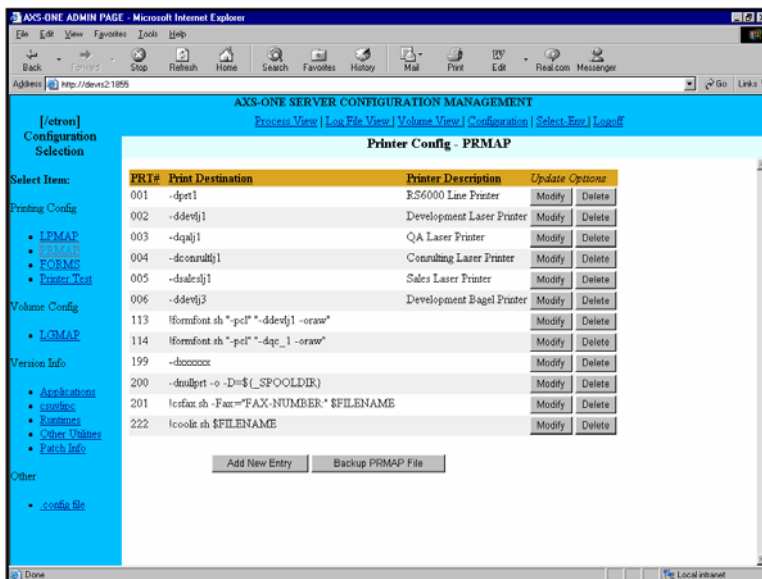
This displays the contents of the LPMAP file. This is similar to editing `$CTRON_EXEC/_ctron_/wisp/config/LPMAP` from the shell. For more information about this configuration file, refer to Appendix A, “Printer Setup.”



Configuration – Printing Configuration LPMAP

PRMAP

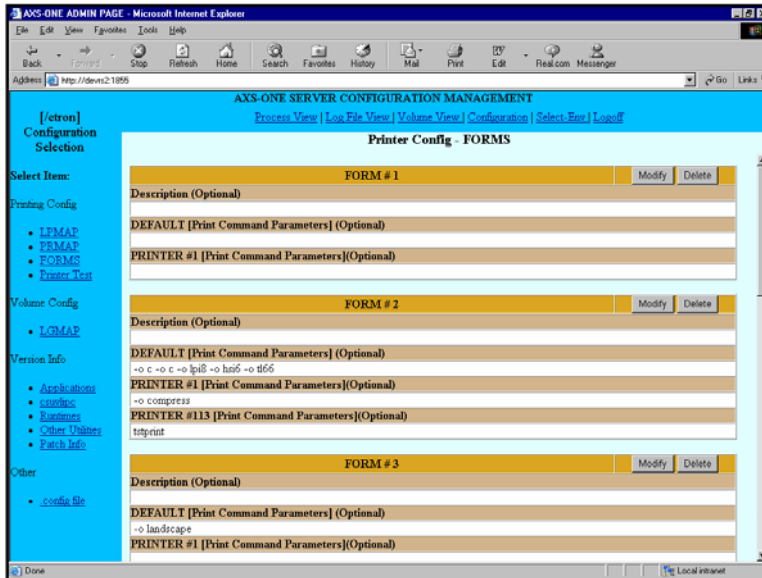
This displays the contents of the PRMAP file. This is similar to editing `$CTRON_EXEC/_ctron_/wisp/config/PRMAP` from the shell. For more information about this configuration file, refer to Appendix A, “Printer Setup.”



Configuration – Printing Configuration PRMAP

FORMS

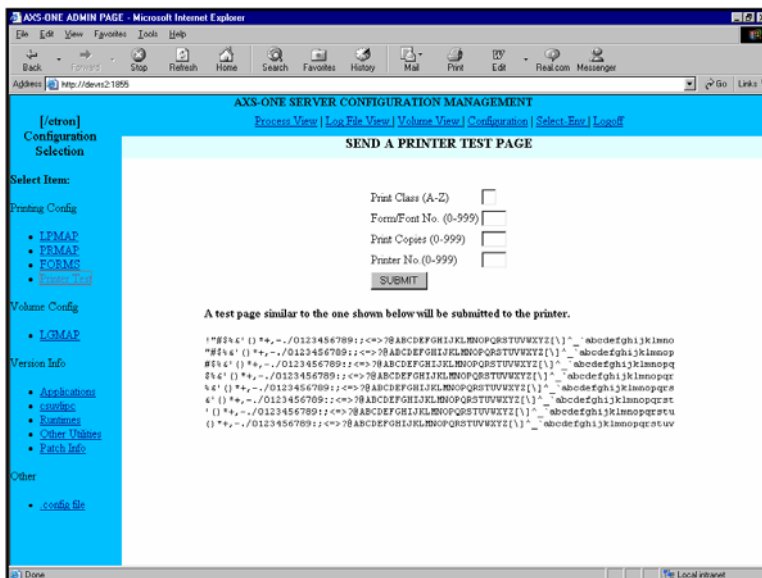
This displays the contents of the FORMS file. This is similar to editing \$CTRON_EXEC/_ctron_/wisp/config/FORMS from the shell. For more information about this configuration file, refer to Appendix A, "Printer Setup."



Configuration – Printing Configuration FORMS

Printer Test

This allows the user to send a test page to the printer. You are prompted for the Printer Class, Form Font, Printer Number and the Number of Copies. For more information about this configuration file, refer to Appendix A, "Printer Setup."

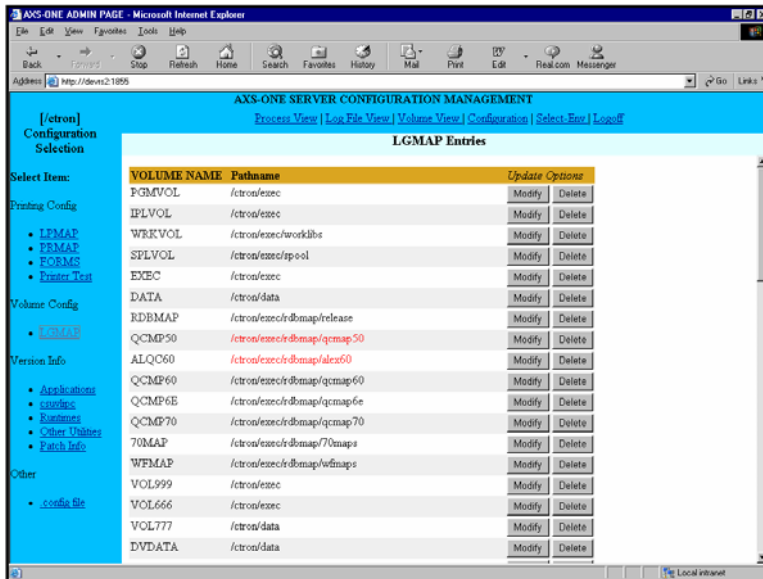


Configuration – Printing Configuration Send a Printer Test Page

Volume Config

LGMAP

This displays the contents of the LGMAP file. This is similar to editing \$CTRON_EXEC/_ctron_/wisp/config/LGMAP from the shell. For more information about this configuration file, refer to Appendix C, “LGMAP File.”



The screenshot shows a web browser window displaying the AXS-ONE SERVER CONFIGURATION MANAGEMENT interface. The main content area is titled "LGMAP Entries" and contains a table with the following data:

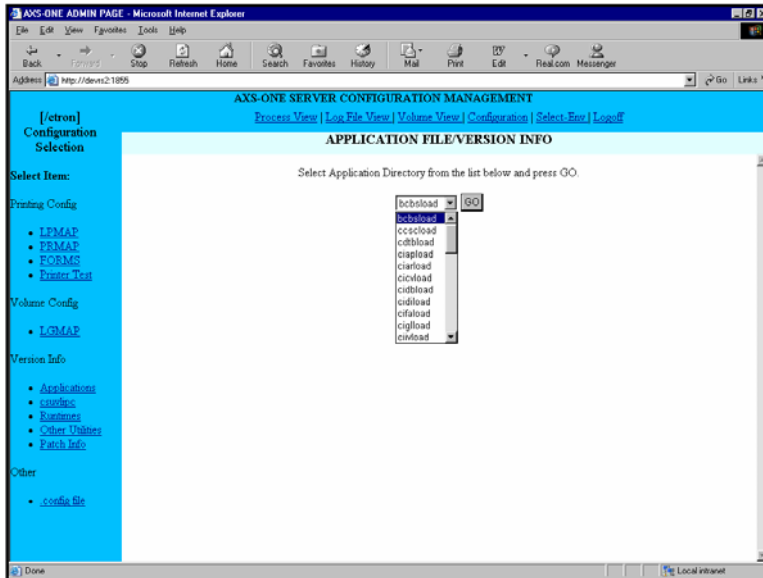
VOLUME NAME	Pathname	Update Options
PGMVOL	/ctron/EXEC	Modify Delete
IPLVOL	/ctron/EXEC	Modify Delete
WRKVOL	/ctron/EXEC/worklib	Modify Delete
SPLVOL	/ctron/EXEC/spool	Modify Delete
EXEC	/ctron/EXEC	Modify Delete
DATA	/ctron/data	Modify Delete
RDBMAP	/ctron/EXEC/rdbmap/release	Modify Delete
QCMP50	/ctron/EXEC/rdbmap/qcmap50	Modify Delete
ALQC60	/ctron/EXEC/rdbmap/alcx60	Modify Delete
QCMP60	/ctron/EXEC/rdbmap/qcmap60	Modify Delete
QCMP6E	/ctron/EXEC/rdbmap/qcmap6e	Modify Delete
QCMP70	/ctron/EXEC/rdbmap/qcmap70	Modify Delete
70MAP	/ctron/EXEC/rdbmap/70maps	Modify Delete
WFEMAP	/ctron/EXEC/rdbmap/wfemaps	Modify Delete
VOL599	/ctron/EXEC	Modify Delete
VOL666	/ctron/EXEC	Modify Delete
VOL777	/ctron/data	Modify Delete
DVDDATA	/ctron/data	Modify Delete

Configuration – Volume Configuration LGMAP

Version Info

Applications

This presents a pull down menu with all of the available applications listed. The desired application is pulled up and the GO button is pushed. This displays the programs within the application along with the version and file information (size, timestamp, etc).



Configuration – Version Information Applications

AXS-ONE SERVER CONFIGURATION MANAGEMENT
 Process View | Log File View | Volume View | Configuration | Select-Env

APPLICATION FILE/VERSION INFO

Select Item: == DIRECTORY CONTENTS OF /ctron/exec/ciapload ==

PERMISSION	LINKS	OWNER	GROUP	FILESIZE	DATE	FILENAME	VERSION#
-rw-rw----	1	ctronsys	ctron	25088	Jul 27 2006	APBATMTI	10.00.00
-rw-rw----	1	ctronsys	ctron	11264	Jul 27 2006	APFAUEXT	10.00.01
-rw-rw----	1	ctronsys	ctron	7168	Jul 27 2006	APGSTUX	10.00.00
-rw-rw----	1	ctronsys	ctron	2816	Jul 27 2006	APTAEXT	10.00.00
-rw-rw----	1	ctronsys	ctron	1280	Jul 27 2006	APTAXEXT	10.00.00
-rw-rw----	1	ctronsys	ctron	11264	Jul 27 2006	APTBUEXT	10.00.00
-rw-rw----	1	ctronsys	ctron	155392	Feb 20 13:22	APUEXT	10.00.01
-rw-rw----	1	ctronsys	ctron	5120	Jul 27 2006	APUEXT2	10.00.00
-rw-rw----	1	ctronsys	ctron	20992	Jul 27 2006	COAPPARM	10.00.00
-rw-rw----	1	ctronsys	ctron	21760	Jul 27 2006	COAPRGHI	10.00.00
-rw-rw----	1	ctronsys	ctron	37632	Jul 27 2006	COBANKI	10.00.00
-rw-rw----	1	ctronsys	ctron	19200	Jul 27 2006	COCHECKI	10.00.00
-rw-rw----	1	ctronsys	ctron	20480	Jul 27 2006	COHISTI	10.00.00

Configuration – Version Information ciapload Version Numbers

csuvlipc

This displays the version of the csuvlipc program, along with the subroutines that make up the csuvlipc program along with size and date information.

The screenshot shows the 'CSUVLIPC INFO' section of the configuration management tool. It displays the following details:

```

-rw-r--r-- 1 ctronsys ctron 9784632 Mar 15 04:54 /ctron/exec/csuvlipc
csuvlipc 10.00.01
/ctron/exec/csuvlipc: object library
VERSLIPC: size: 255 (FF), v4.1, VAX-mode (07-Jul-2006)
GETUSER: size: 24414 (5F5E), v4.1, VAX-mode, debug (17-Jan-2006)
GETAFIL: size: 806 (326), v2.3, VAX-mode, resident (11-Dec-2000)
CSMENUIC: size: 25476 (6384), v4.1, VAX-mode, debug (28-May-2003)
CSMENUIC: size: 15313 (3BD1), v2.3, VAX-mode, resident (11-Dec-2000)
SCNDRVR: size: 252555 (3DA8B), v4.1, VAX-mode, resident, debug (27-Apr-2006)
SCNDRVRC: size: 161332 (27634), v2.3, VAX-mode, resident, debug (03-Jul-2001)
SCNFILL: size: 408 (198), v2.3, VAX-mode, resident (11-Dec-2000)
CSHAPCD: size: 43031 (A817), v4.1, VAX-mode, resident (25-Apr-2006)
CSTTDEP: size: 13793 (35A5), v4.1, VAX-mode, resident (25-Aug-2003)
CTLVAL2: size: 88275 (158D3), v4.1, VAX-mode, resident (28-Jan-2003)
GETALOC: size: 13933 (366D), v4.1, VAX-mode (12-Apr-2005)
INQZ: size: 95320 (17458), v4.1, VAX-mode, resident, debug (06-Jul-2006)
PUTUSER: size: 4381 (111D), v4.1, VAX-mode (16-Nov-2001)
QMARK2: size: 56131 (DB43), v4.1, VAX-mode, resident, debug (07-Nov-2005)
ULNDRVR: size: 46802 (B6D2), v4.1, VAX-mode, resident, debug (27-Apr-2006)
SCNINQFM: size: 32452 (7EC4), v4.1, VAX-mode, resident, debug (07-Jul-2006)
PRINTCRT: size: 6790 (1A86), v2.3, VAX-mode (11-Dec-2000)
  
```

Configuration – Version Information csuvlipc Information

Runtimes

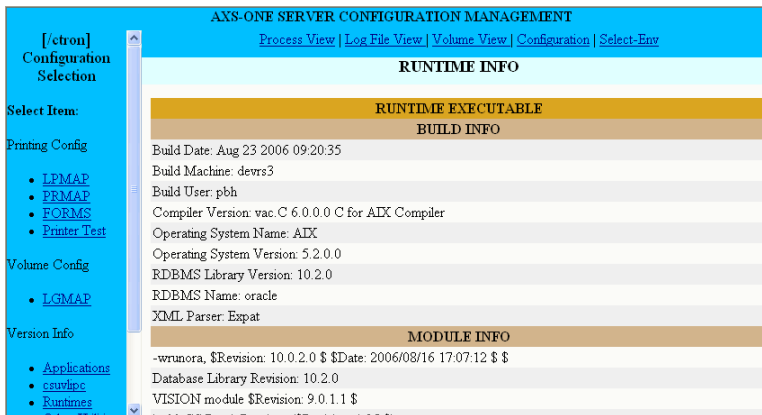
This displays the available runtimes in the \$CTRON_EXEC/_ctron_ directory, the version, and the date of compile.

The screenshot shows the 'RUNTIME INFO' section of the configuration management tool. It displays a table of runtime executables and their version information:

RUNTIME EXECUTABLE	VERSION INFO
wruncl2	\$\$Revision: 10.0.1.0 \$ \$Date: 2006/09/27 18:46:18 \$
wruncl2.old	\$\$Revision: 10.0.0.1D \$ \$Date: 2006/09/08 12:25:01 \$ \$
wrunora	\$\$Revision: 10.0.3.0D \$ \$Date: 2007/23/03 00:07:22 \$ \$
wrunora.10.0.2.0.10.2.0	\$\$Revision: 10.0.2.0 \$ \$Date: 2006/08/16 17:07:12 \$ \$
wrunora.10.0.2.0.9.2.0	\$\$Revision: 10.0.2.0 \$ \$Date: 2006/08/16 17:07:12 \$ \$
wrunora.10.0.3.0.9.2.0	\$\$Revision: 10.0.3.0 \$ \$Date: 2007/03/15 15:46:05 \$ \$
wrunora.10.0.3.0.9.2.0.old	\$\$Revision: 10.0.3.0 \$ \$Date: 2007/03/15 15:46:05 \$ \$
wrunora.10.0.4.0.10.2.0	\$\$Revision: 10.0.4.0 \$ \$Date: 2007/03/23 20:31:39 \$ \$
wrunora.10.0.4.0.9.2.0	\$\$Revision: 10.0.4.0 \$ \$Date: 2007/03/23 20:31:39 \$ \$
wrunora.10.0.5.0.10.2.0	\$\$Revision: 10.0.5.0 \$ \$Date: 2007/03/29 19:34:37 \$ \$
wrunora.10.0.5.0.9.2.0	\$\$Revision: 10.0.5.0 \$ \$Date: 2007/03/29 19:34:37 \$ \$
wrunora.10.0.6.0.10.2.0	\$\$Revision: 10.0.6.0 \$ \$Date: 2007/03/30 13:38:35 \$ \$
wrunora.10.0.6.0.9.2.0	\$\$Revision: 10.0.6.0 \$ \$Date: 2007/03/30 13:38:35 \$ \$
wrunora.10.0.7.0.10.2.0	\$\$Revision: 10.0.7.0 \$ \$Date: 2007/04/10 19:34:09 \$ \$

Configuration – Version Information Runtime Information

When you click on the **RUNTIME EXECUTABLE** link, you are presented with more details about the runtime.

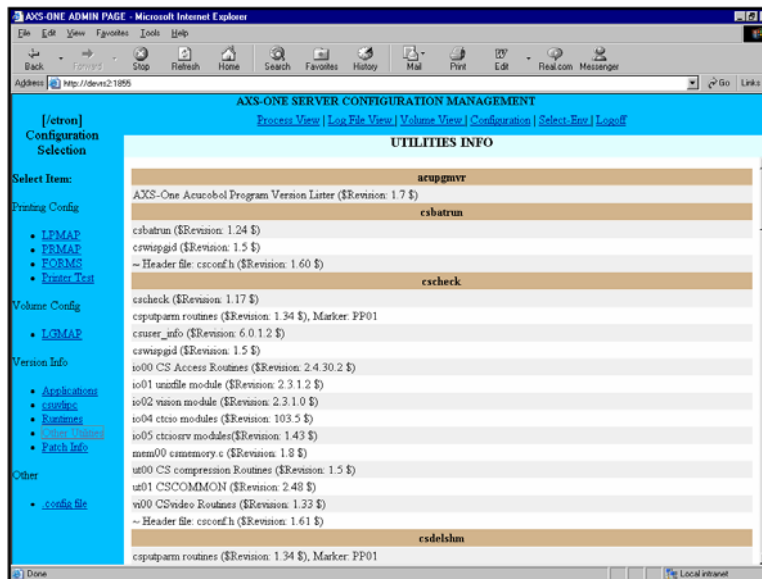


Configuration – Version Information Runtime Executable Information

Other Utilities

This displays the various utilities and their version numbers.

Note: Depending on the speed of the server, this can take a lot of time and the browser may time out before the page is displayed.



Configuration – Version Information Other Utilities

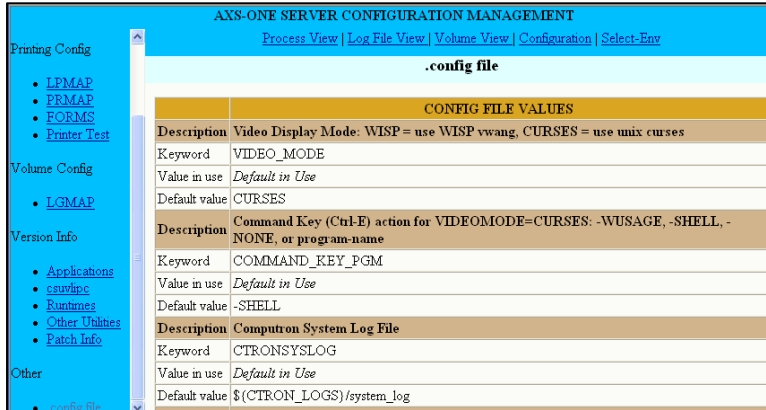
Patch Info

This displays the patch release information for the environment, including the Product, the date of the full install, and the date of the last patch. This is similar to editing the `$CTRON_ROOT/.last_patch` file from the shell.

Other

.config

This displays the contents of the .config file. This is similar to editing the \$CTRON_EXEC/_ctron_/.config file from the shell.



The screenshot shows a web-based configuration interface titled "AXS-ONE SERVER CONFIGURATION MANAGEMENT". The main content area is titled ".config file" and displays a table of configuration values. The table has columns for Description, Keyword, Value in use, and Default value. The interface includes a left-hand navigation menu with categories like "Printing Config", "Volume Config", "Version Info", and "Other".

CONFIG FILE VALUES			
Description	Video Display Mode: WISP = use WISP vvwang, CURSES = use unix curses		
Keyword	VIDEO_MODE	Value in use	Default in Use
		Default value	CURSES
Description	Command Key (Ctrl-E) action for VIDEOMODE=CURSES: -WUSAGE, -SHELL, -NONE, or program-name		
Keyword	COMMAND_KEY_PGM	Value in use	Default in Use
		Default value	-SHELL
Description	Computron System Log File		
Keyword	CTRONSYSLOG	Value in use	Default in Use
		Default value	\$(CTRON_LOGS)/system_log

Configuration – Other .config file

5.2 Batch Processing Setup for Unix - csenq

Introduction to Batch Processing Setup for Unix – csenq

In Computron versions 7.0 and higher, the utility that manages background processing, is known as csenq. Computron provides three ways to run jobs and reports. A process can be run in Screen mode, which displays the first page full of data as soon as it is retrieved. It can be run in Foreground, which displays a “Processing in Progress” message in the window until completion and produces a report. It can be run in background, which leaves the user free to do other things in the active window and produces a report.

This utility, csenq, replaces the lp based processing which was used in the past. Directions for managing lp based systems can be found in Appendix E.

Benefits

- ◆ Greater flexibility
- ◆ Multiple background queues by default
- ◆ Reduced administration
- ◆ Store queue names from A to Z for background jobs
- ◆ Ability to see additional information on each of the queues and the jobs within each of the queues
- ◆ Ability to cancel background jobs, move jobs between queues, hold jobs for deferred execution, as well as job status logging.

Installation of csenq

The background queue is normally setup during the installation process. However, modifications to the Computron environment can be accomplished by executing the program, ctron_install, located in the Computron root, .../exec/_ctron_, directory. Only an individual authorized to act as root can execute the ctron_install command.

```
3 - Install Computron Batch (lp or csenq) queue
```

```
This option installs a batch queue for running background jobs.
```

```
Select background queue management system: (c)senq, (l)pqueue or (q)uit
(enter c, l or q)?
```

csenq

The following csenq commands are available at the Unix command line:

- ◆ csenq -l [-v] [QueueNameList]
- ◆ csenq -c job1 [job2 [job3 ...]]
- ◆ csenq -h QueueNameList
- ◆ csenq -r QueueNameList

- ◆ `cseq -a job1 [job2 [job3 ...]]`
- ◆ `cseq -m QueueName job1 [job2 [job3 ...]]`

The following are `cseq` command modifiers and definitions:

- l – List the jobs in the queue.
 - c – Cancel a job or jobs.
 - h – Hold a queue.
 - r – Release a queue.
 - v – Verbose output.
 - a – Add an incomplete job back to a queue.
 - m – Move a job from one queue to another. This does not apply to jobs in progress.
- QueueName – This is a letter between A and Z.
- QueueNameList – This is QueueNames concatenated (e.g. ABC).
- job# - This is a numeric value of a job (from `cseq -l`).

Sample Output of the `cseq -lv` Command

```
CSENQD status:
  Log File =
  Logging Level = 1
  Group Init = ctron
  Admin List = root,ctronsys
Queue=A State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=B State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=C State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=D State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=E State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=F State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
```

The *CSENQD.CONF* file

This file controls the variables that are passed to the `cseqd` daemon. The `cseqd` daemon is started automatically with the first submission of a job to background processing from within a Computron application. This file is owned by ROOT and may only be modified by an individual with ROOT authority.

Permissions of file:

```
-rw----- 1 root    ctron      1713 Oct 04 08:07 cseqd.conf
```

Sample file contents

```

# csenqd.conf -- created Thu Oct  4 08:07:43 2006
# To provide csenqd with variables to work with

# LOGGING_LEVEL determines the verbosity of the logfile.
# LOGGING_LEVEL=0 (no log).
# LOGGING_LEVEL=1 (default; logs start,submitted,run & end.)
# LOGGING_LEVEL=2 (debug).
LOGGING_LEVEL=1
# LOGFILE=

# GROUP_INIT determines what setgroups() should be set to.
# GROUP_INIT=group1,group2,...
GROUP_INIT=ctron

# ADMIN_LIST determines who have authority to execute special tasks.
# ADMIN_LIST=admin1,admin2,...
ADMIN_LIST=ctronsys

# NICE sets the nice value of the individual queue.
# MAXJOBS sets the maximum jobs that can be run in the queue.
# STATE sets the queue to either: E = Enabled, H= Held.
QUEUE=A, NICE=5, MAXRUNJOBS=3, STATE=E
QUEUE=B, NICE=5, MAXRUNJOBS=3, STATE=E
QUEUE=C, NICE=5, MAXRUNJOBS=3, STATE=E

```

Definition of entries in file

LOGGING_LEVEL

This determines the verbosity of the log file. Valid values are:

0 – No log is generated.

1 –This logs the start, submitted, run & end of every batch process. This is the default value.

2 – Debug captures a large amount of data. Remember to turn it off when the issue is resolved to prevent massive amounts of information being captured.

LOGFILE

Do not enter information here unless you desire to have the logfile generated to a location other than the default for logs (COMPUTRON_ROOT/var).

GROUP_INIT

This determines what setgroups() should be set to

```
GROUP_INIT=group1 , group2 ,
```

If for some security reason you don't use the group "ctron", substitute or add another group here. Separate entries with commas; do not use spaces between entries.

ADMIN_LIST

This determines who has authority to execute special tasks.

```
ADMIN_LIST=admin1 , admin2 , ...
```

The defaults are "root" and "ctronsys". If you wish to add additional members to this list do so with caution. Separate entries with commas; do not use spaces between entries. Members must be authorized to Unix and be in the Computron user file.

STATE

This sets the queue to either: E = Enabled or H= Held. You may set a specific queue to a held state. You might consider changing this parameter if you desire to have certain jobs not execute in an automatic fashion. It allows you to move a specific held job from the held queue to an enabled queue for processing. To have specific jobs run to a background queue other than "A" you must use a character cell session of Computron. Once logged into the Unix Computron Financial system, you can have this session output background jobs to another background queue by selecting PF-25. Enter the queue change into the Queue Name field. See the window under the heading, " Sample DS1 Window Displayed by PF – 25."

MAXJOBS

This sets the maximum jobs that can be run in the queue. The default is (3) three. This can be compared to having 3 background queues using the Unix lp spooler. You can increase or decrease this parameter. Increasing the parameter can negatively impact overall performance of your Unix system and should only be changed if sufficient system resources are available. Setting the parameter to (0) effectively shuts down the queue

Sample DS1 Window Displayed by PF - 25

```

Telnet supsun2
--- Set Defaults for User ---          DS1
Enter Indicated Data and Press <RETURN>
--- or Press <PF-16> to Exit

Data Base Map:  LBPK10G              in  CSDBMAP              on  DATA
Default Company      Company scnio:      Batch Stream:
I/O Diagnostics     N  <Off/Summary/Full>

Work/temporary files:      WORKDIR  WK024227      WORKUOL  WRKUOL
User programs:            RUNDIR   RUNUOL      EXEC

--- For Submitted Jobs ---

Run or Hold  R (R/H)      Priority:  0
Queue Name:

RUNTIME=/enterprise/10qa/exec/_ctron_/wrunch12,  REU=10.0D
MISPGID=24227,  PID=24479,  NODENAME=supsun2,  OS=SunOS<5.10>

```

Set Defaults for User Window in a Character Cell session

Sample log file of job running in queue B

```

Queue=A State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=B State=enabled Jobs=[Running=1, Waiting=0, Max=3] Nice=5
JOB #   USER  PROCESS-ID  START-DATE-TIME  STATUS
OPENFILE OPENLIB  OPENVOL  PROGRAM  PROGLIB  PROGVOL  DESCRIPTION
-----
21      amf      12874      10/10/01 12:20:46  Running
WHMAPSV CSDBMAP  DATA    RPT1000B CIGLOAD EXEC    Trial Balance Report

```

Sample log file of job running in queue A

```

Queue=A State=enabled Jobs=[Running=3, Waiting=1, Max=3] Nice=5
JOB #   USER  PROCESS-ID  START-DATE-TIME  STATUS
OPENFILE OPENLIB  OPENVOL  PROGRAM  PROGLIB  PROGVOL  DESCRIPTION
-----
22      amf      18952      10/10/01 13:24:53  Running
WHMAPSV CSDBMAP  DATA    RPT1000B CIGLOAD EXEC    Trial Balance Report
-----
23      amf      17904      10/10/01 13:24:57  Running
WHMAPSV CSDBMAP  DATA    RPT1000B CIGLOAD EXEC    Trial Balance Report
-----
24      amf      21952      10/10/01 13:24:59  Running
WHMAPSV CSDBMAP  DATA    RPT1000B CIGLOAD EXEC    Trial Balance Report
-----
25      amf      10/10/01 13:25:01  Waiting
WHMAPSV CSDBMAP  DATA    RPT1000B CIGLOAD EXEC    Trial Balance Report
-----
Queue=B State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=C State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5

```


5.3 Analyzer

Introduction to the Analyzer

The Analyzer is a program that takes system usage data and presents it in a useful fashion. It can be presented in tables, graphs or charts and it can also be exported to a text file or to an XML file for use in other applications.

Client software must be installed on a Windows2000, Windows NT or Windows 2003 machine.

Setup

To collect statistical data for the Analyzer, the following entry must be made in the .config file in `.../exec/_ctron_`

```
COLLECT_STATS=YES
```

The raw data is stored in a file called CSMasterStat.Dat located in the `.../var/Stat` directory. Without this entry in the .config file, no data is collected.

Converting the raw data file

The raw data in the CSMasterStat.Dat file must be converted to a Microsoft Access database to use with the Analyzer. You can do this in one of two ways:

- ◆ Convert the data using CollectStat on the Command line
- ◆ Open the Analyzer and import the data into the database.

Both of these methods are described below.

Using CollectStat on the Command Line

1. Move the CSMasterStat.Dat file from the Unix server to the client machine.
2. Execute the following command on the client machine to convert the data in the CSMasterStat.Dat file into a Microsoft Access database format:

```
CollectStat -i"c:\ctron\var\stat\CSMasterStat.Dat"  
-d".../exec/stat/test1.mdb"
```

☞ *Note:* -i is the original .dat file, -d is the output database.

☞ *Note:* The path can be omitted if the files are in the current directory, so another valid command line would be:

```
CollectStat -i"CSMasterStat.Dat" -d"test1.mdb"
```

☞ *Note:* The CollectStat program cleans out the original .dat file. That means that any information that was in the original .dat file is now only available in the .mdb file it was

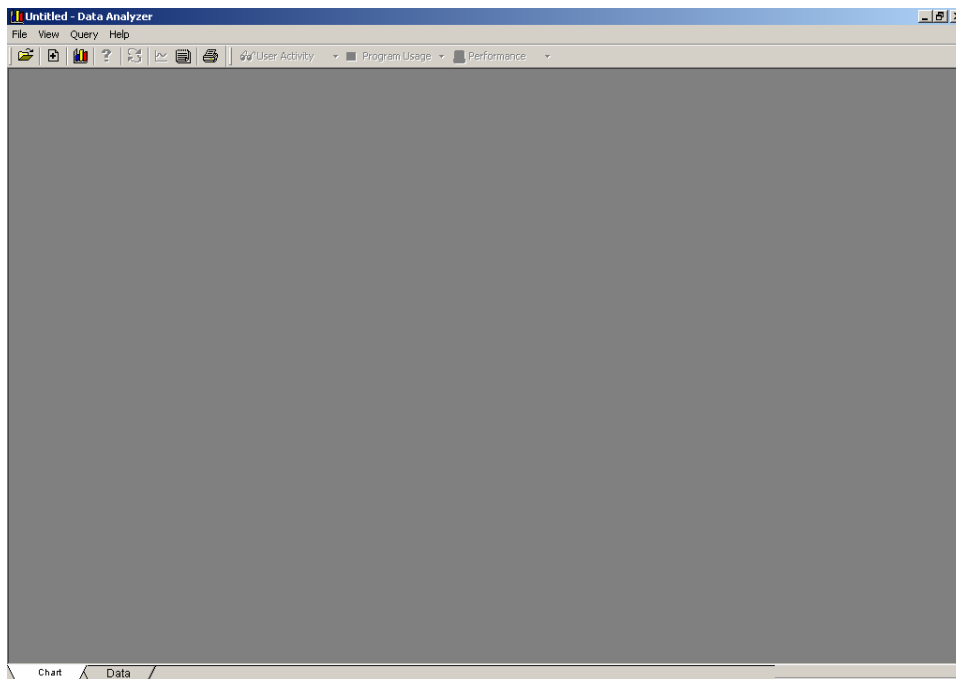
converted to. CollectStat does not delete the CSMasterStat.dat file, but it starts collecting data again.

- ☞ *Note:* If you experience errors importing the data into the Analyzer database, it may be necessary to install the latest version of MDAC (the Microsoft Data Access Components) on your system. See “Troubleshooting” later in this section.

Using the Analyzer

Displaying new data

To run the Analyzer, execute the Analyzer.exe program.



Analyzer initial screen

Using an already converted data file

- ☞ *Note:* If the raw data file is not already converted (i.e., you did not execute the CollectStat statement), refer to the instructions below the heading “Using a data file not already converted.”

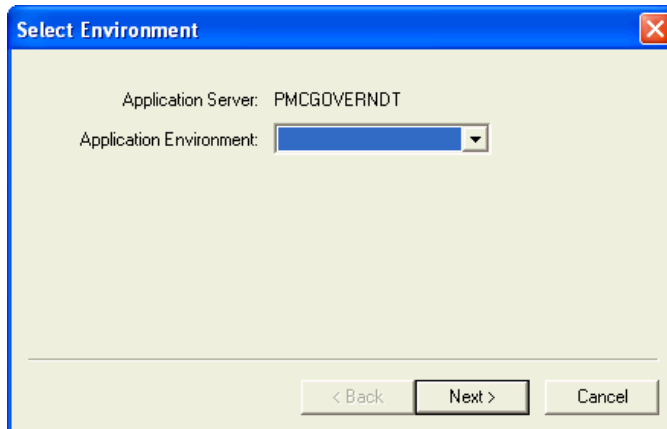
1. Select the Open command from the File menu.
2. Select the Access database (.mdb file) that you would like to open.

- ☞ *Note:* The Access database file is the raw data file, converted in a previous step.

- ☞ *Note:* Depending on how you have been collecting your data, the MDB files can have a different naming convention and/or location.

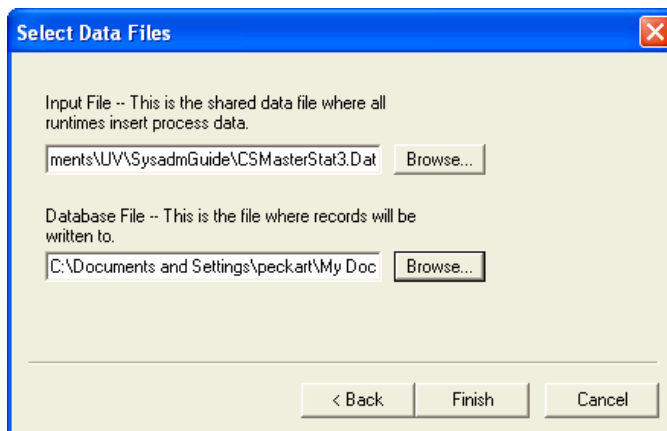
Using a data file not already converted

1. Select Import New Data from the File menu, to import data from the local Computron application server.
2. Select the environment where the Analyzer data was collected. You can leave this field blank.



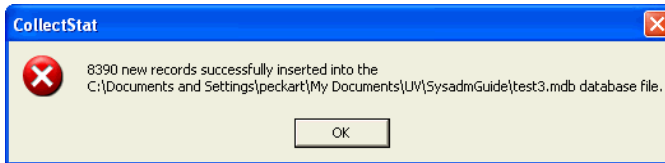
Import Data Select Environment Window

3. Click Next
4. Browse (or enter) the path to the raw data file. Then browse to or enter the path where the converted data file should be placed.



Import Data Select Data Files Window

5. Click Finish
6. A message box appears telling you how many records were imported to the database file.



Import Data Message when new data is imported

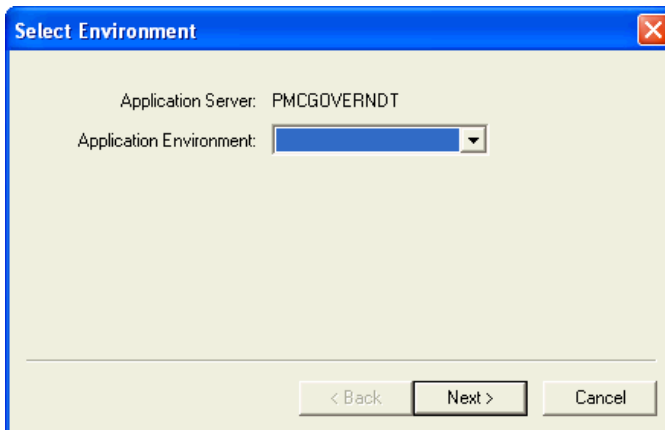
7. If no other file is currently loaded in the Analyzer, the database file just created is automatically opened.

☞ *Note:* This process runs the CollectStat program which cleans out the original .dat file. That means that any information that was in the original .dat file is now only available in the .mdb file it was converted to. CollectStat does not delete the CSMasterStat.dat file, but it starts collecting data again.

☞ *Note:* If there are errors importing the data into the Analyzer database, it may be necessary to install the latest version of MDAC (the Microsoft Data Access Components) on your system. See “Troubleshooting” later in this section.

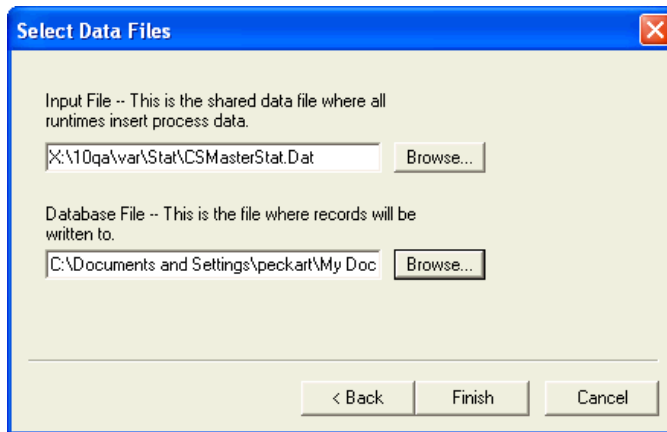
Adding data to an already existing database file

1. Run the Analyzer, by executing Analyzer.exe.
2. Open the already created database file by selecting Open from the File menu and browsing to the desired database file.
3. Select the Import New Data option from the File menu.
4. Select the environment where the Analyzer data was collected. You can leave this field blank.



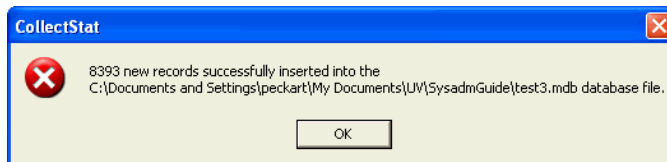
Import New Data Select Environment Window

5. Click Next.
6. Verify (and correct, if necessary) the path to the raw data file and the path to the currently open database file.



Import New Data Select Data Files Window

7. Click Finish.
8. A message box appears telling you how many records were imported to the database file.



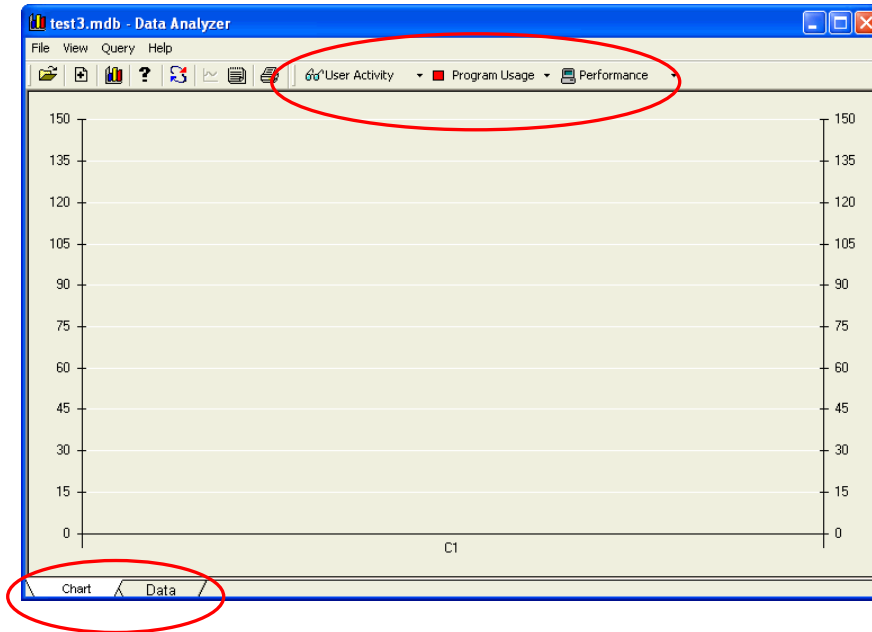
Import New Data Message when data is added to an existing database

9. Click the “Refresh” button on the current display. This automatically incorporates any new data that has been imported.

Note: This process runs the CollectStat program which cleans out the original .dat file. That means that any information that was in the original .dat file is now only available in the .mdb file it was converted to. CollectStat does not delete the CSMasterStat.dat file, but it starts collecting data again.

Analyzing the data

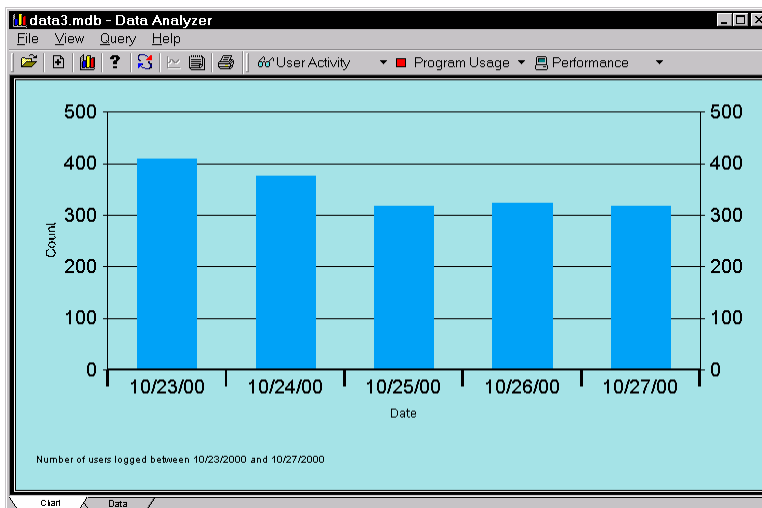
The data is organized and displayed by selecting one of the analysis options from the User Activity, Program Usage or Performance menus.



Note: For most options, the display defaults to the Chart view. However, to see the Data view, simply click on the Data tab on the bottom of the Analyzer window.

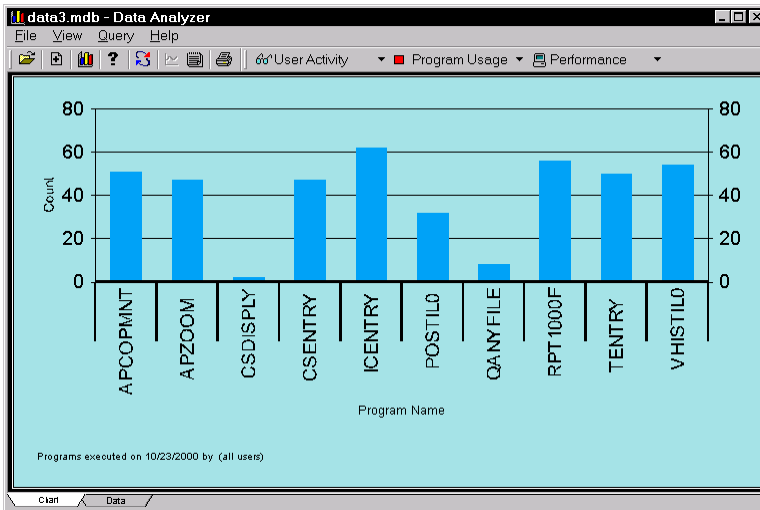
User Activity Menu:

- ◆ **Users logged on each day from x to y date** - This produces a graph of the total number of user logged into the system each day within a selected time period.



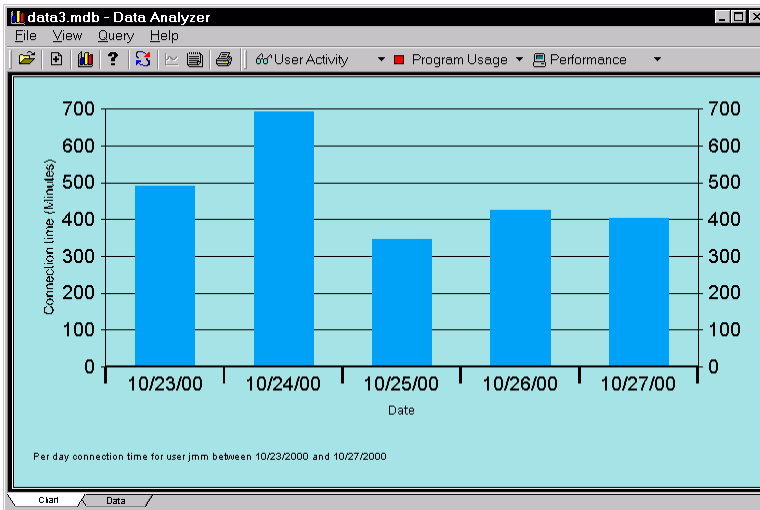
Total users logged on in a given date range

- ◆ **User Activity (programs executed) on a particular day** - This produces a graph of programs executed on a particular day. Choose to see all users or filter it to a particular user.



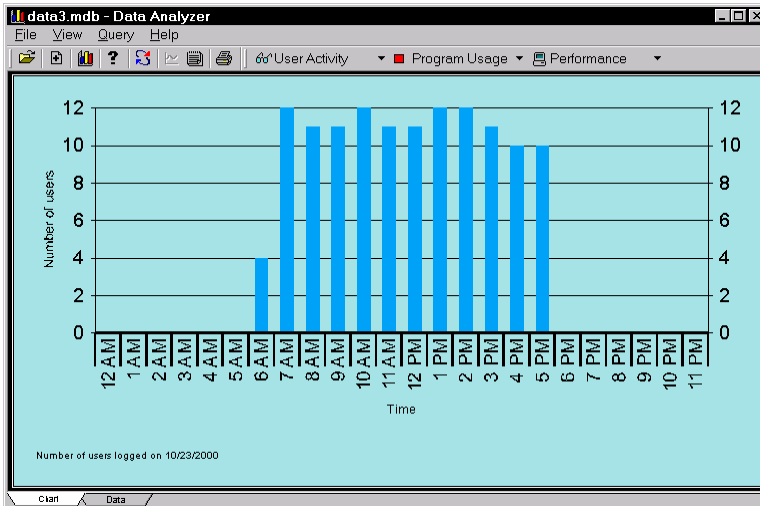
Programs executed on a particular day

- ◆ **Total connection time per user from x to y date** - This shows the total time (in minutes) that users are connected to the system each day. It is possible to choose to see a single user, or the total time for all users.



Total connection time for users in a given date range

- ◆ **Number of users logged in on a particular day** - This produces a timeline output to see how many users are logged on during various times of a certain day.



Number of users logged in during a day

- ◆ **User Activity – Detailed** — Returns detailed data about programs that have been run. This output can be filtered by user and time period.

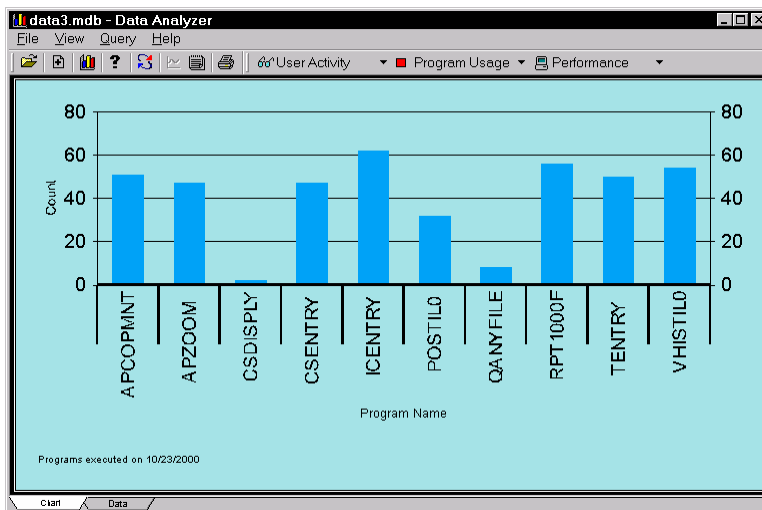
☞ *Note:* There is no “Chart View” associated with this query

UserName	ProgramNa...	StartDate	StartTime	EndTime
jmm	CSENTRY	10/23/00	7:16:00 AM	7:49:00 AM
jmm	APZOOM	10/23/00	7:26:00 AM	7:29:00 AM
jmm	TENTRY	10/23/00	7:32:00 AM	8:26:00 AM
jmm	VHISTILO	10/23/00	7:32:00 AM	7:43:00 AM
jmm	APCOPMNT	10/23/00	8:09:00 AM	8:24:00 AM
jmm	TENTRY	10/23/00	8:13:00 AM	8:33:00 AM
jmm	CSENTRY	10/23/00	8:31:00 AM	8:32:00 AM
jmm	TENTRY	10/23/00	8:45:00 AM	9:01:00 AM
jmm	APCOPMNT	10/23/00	9:02:00 AM	9:40:00 AM
jmm	TENTRY	10/23/00	9:11:00 AM	9:32:00 AM
jmm	APZOOM	10/23/00	9:19:00 AM	9:45:00 AM
jmm	VHISTILO	10/23/00	9:30:00 AM	10:02:00 AM
jmm	CSENTRY	10/23/00	9:41:00 AM	10:24:00 AM
jmm	ICENTRY	10/23/00	9:42:00 AM	10:25:00 AM
jmm	RPT1000F	10/23/00	10:40:00 AM	10:44:00 AM
jmm	APZOOM	10/23/00	10:41:00 AM	10:54:00 AM
jmm	TENTRY	10/23/00	10:44:00 AM	10:52:00 AM
jmm	ICENTRY	10/23/00	11:28:00 AM	11:46:00 AM
jmm	APCOPMNT	10/23/00	11:44:00 AM	11:51:00 AM
jmm	TENTRY	10/23/00	12:02:00 PM	12:17:00 PM
jmm	CSENTRY	10/23/00	1:34:00 PM	1:37:00 PM
jmm	RPT1000F	10/23/00	2:06:00 PM	2:20:00 PM
jmm	TENTRY	10/23/00	2:07:00 PM	2:14:00 PM

Detail of user activity

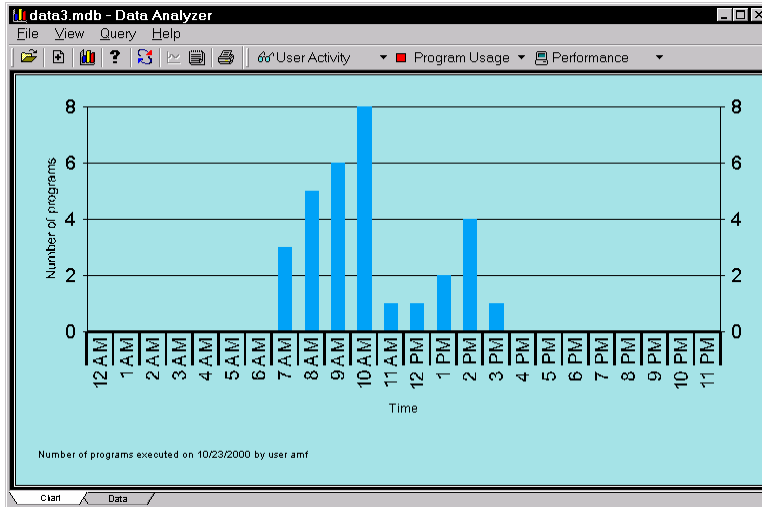
Program Usage Menu:

- ◆ **Distinct programs executed on a particular day** - This option graphs all of the programs executed on a given day (against the number of times each program was executed).



List of programs executed on a given day

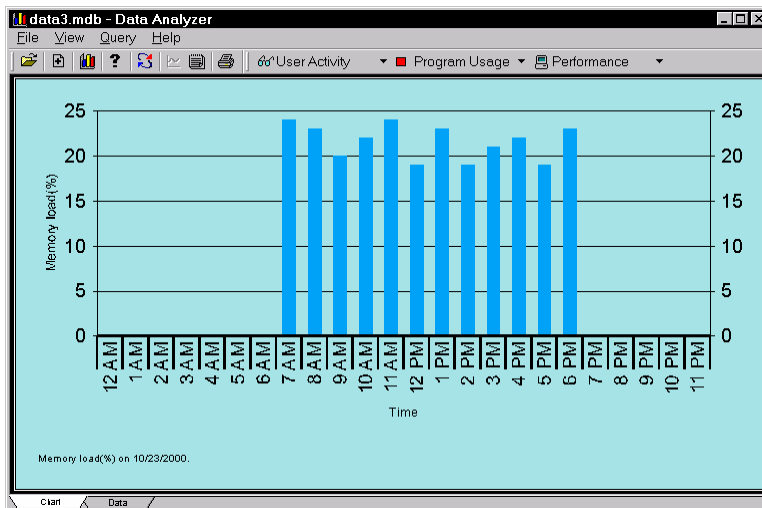
- ◆ **Programs executed vs. time on a particular day** - This output is a timeline (similar to “Number of Users Logged On” above) but it allows filtering by a particular user (shown here) or “all users”. This shows how many programs have been run during the time periods shown.



Programs executed by time

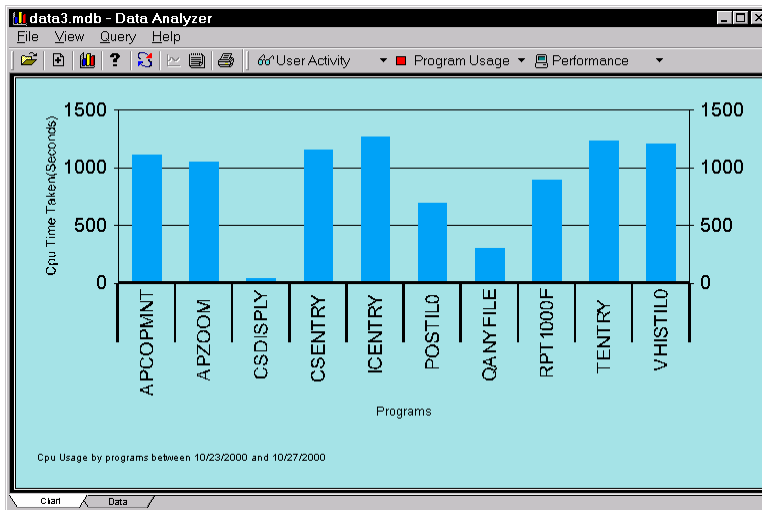
Performance Menu:

- ◆ **Memory Load** - This tracks the memory load on the system (in percent) over the course of a single day. This can be used to isolate memory usage spikes or to see if in general more memory is needed on the system.



Memory usage by day

- ◆ **CPU time used by different programs** – This tracks the actual CPU usage of programs that have been executed over a particular time period. The usage is shown in CPU seconds. This can be used to determine if any one program in particular is being a “CPU hog”.



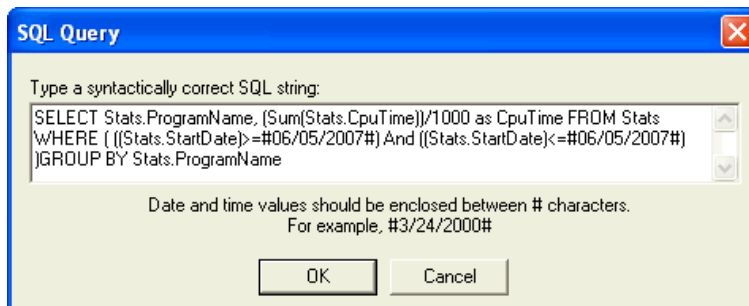
CPU usage by program

Using SQL Queries:

☞ *Note:* It is possible to formulate a query to get specific data from the Analyzer database.

Choose User SQL Command... from the Query menu.

Enter the desired SQL query.



Sample SQL query

The query in the dialog box when it opens is typically the last SQL statement sent to the database.

Click OK to execute the query.

☞ *Note:* Due to the endless combinations of SQL statements, the data is only returned to you in the Data View format (the grid). A Chart View is not available.

Viewing Options:

Chart Properties

Chart properties can be used to modify the look of the charts and graphs. Using the Chart properties, it is possible to change colors, type of chart, fonts, etc.

From the View menu select Chart Properties....

Select the Chart tab to modify a desired property.

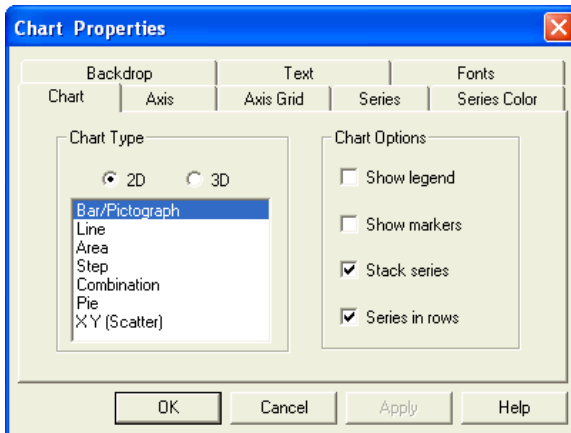
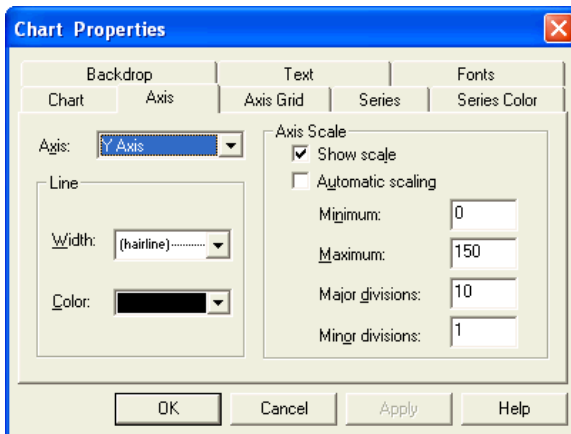


Chart Properties

To change the vertical scale of the chart, do the following:

1. Select the Axis tab.
2. In the Axis field, select Y Axis.
3. Clear the Automatic scaling check box.



4. Set Maximum to the number representing the height of the scale, the highest amount to display.
5. Set the Major divisions to the number of divisions you want to see in the scale.

For example, if you have a scale of 150 and you want a marker every 15, enter 10 in Major divisions. You have a Y axis with marks at 150, 135, 120, 105, 90, etc.

Options...

Using the Options dialog, the data in the chart can be sorted in ascending or descending order.

From the View menu select Options....

Select the order that the data should be sorted.

Note: The ascending/descending sort only applies to certain charts.

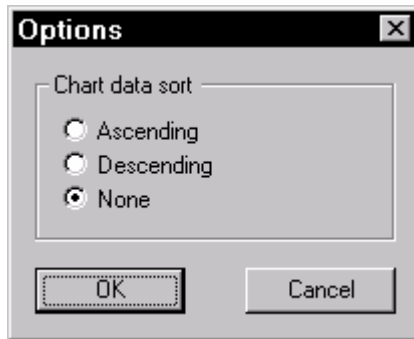
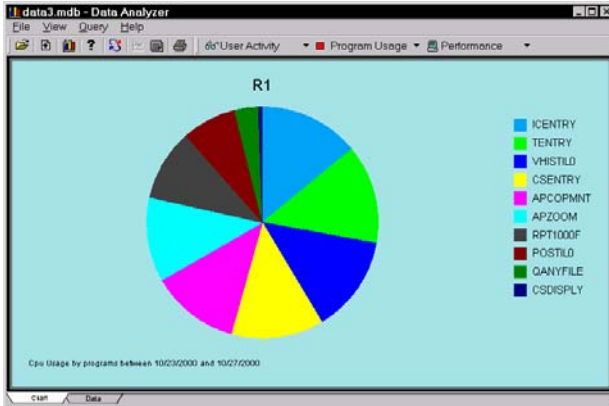
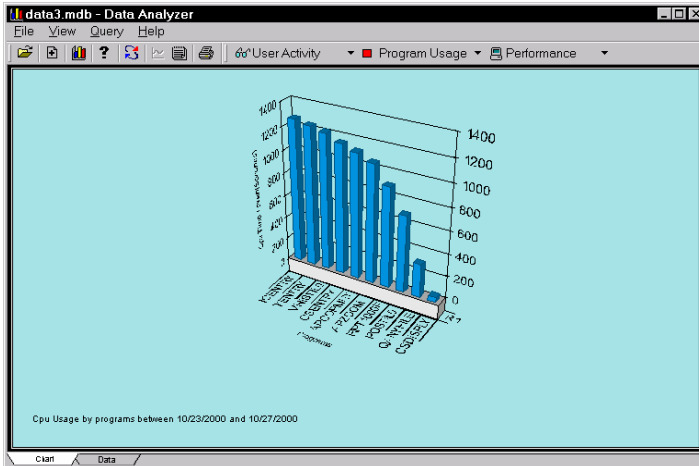


Chart sorting options

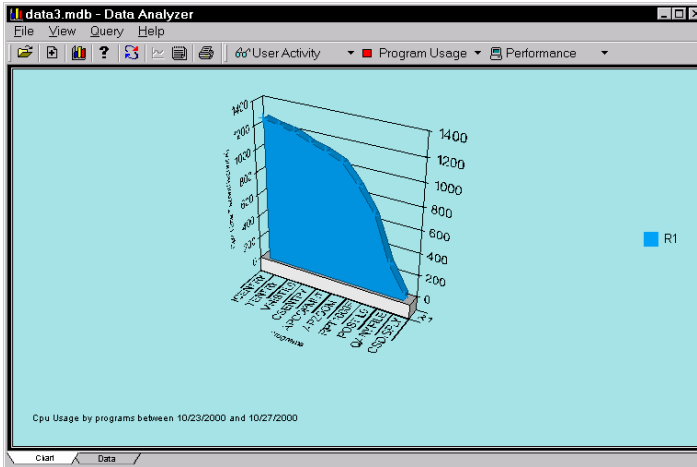
Example Charts



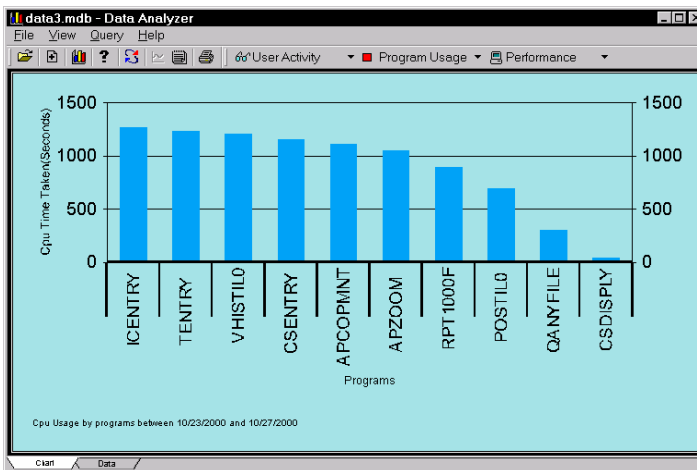
Pie Chart



3-D Chart



3-D Area Chart



Descending order chart

Exporting Data

1. On the File menu, highlight Export Data To.
2. Select Text File or XML File.

Note: Both of these methods export all of the data currently in the Analyzer database. The output is created in the chosen format.

3. Select the path and file name where the exported data is to be placed.

Sample output of a data file exported to Text

The order of entries in this file are:


```
cs,FORMINIT,20357,20357,0,19,0,4/13/2007,3:55:35 PM,4/13/2007,3:55:35 PM
cs,CSENTRY ,20357,20357,0,89,0,4/13/2007,3:55:44 PM,4/13/2007,3:55:48 PM
cs,CSENTRY ,20357,20357,0,89,0,4/13/2007,3:55:44 PM,4/13/2007,3:55:49 PM
```

```
cs,CSENTRY ,20429,20429,0,89,0,4/13/2007,3:56:22 PM,4/13/2007,3:56:28 PM
cs,CSENTRY ,20429,20429,0,89,0,4/13/2007,3:56:22 PM,4/13/2007,3:56:28 PM
ctr,EPICMP ,19089,19089,0,770,0,4/13/2007,2:58:25 PM,4/13/2007,4:50:34 PM
pse,FORMINT,22760,22760,0,29,0,4/16/2007,8:48:16 AM,4/16/2007,8:48:16 AM
pse,REPPASS ,22760,22783,0,259,0,4/16/2007,8:48:55 AM,4/16/2007,8:48:57 AM
pse,AUDUVPR0,22760,22777,0,59,0,4/16/2007,8:48:49 AM,4/16/2007,8:48:57 AM
pse,FORMINT,22760,22760,0,29,0,4/16/2007,8:48:16 AM,4/16/2007,8:48:57 AM
pse,USERMNT3,22760,22760,0,99,0,4/16/2007,8:48:59 AM,4/16/2007,8:50:37 AM
pse,CSEPIC0 ,22760,22791,0,959,0,4/16/2007,8:50:39 AM,4/16/2007,8:51:03 AM
pse,USERMNT3,22760,22760,0,99,0,4/16/2007,8:48:59 AM,4/16/2007,8:51:03 AM
pse,REPPASS ,22760,22822,0,379,0,4/16/2007,8:52:26 AM,4/16/2007,8:52:27 AM
pse,AUDUVPR0,22760,22816,0,49,0,4/16/2007,8:52:21 AM,4/16/2007,8:52:27 AM
pse,USERMNT3,22760,22760,0,109,0,4/16/2007,8:48:59 AM,4/16/2007,8:52:27 AM
pse,REPPASS ,22760,22837,0,749,0,4/16/2007,8:52:34 AM,4/16/2007,8:52:36 AM
pse,AUDUVPR0,22760,22831,0,49,0,4/16/2007,8:52:30 AM,4/16/2007,8:52:36 AM
pse,USERMNT3,22760,22760,0,120,0,4/16/2007,8:48:59 AM,4/16/2007,8:52:36 AM
pse,REPPASS ,22760,22852,0,200,0,4/16/2007,8:52:42 AM,4/16/2007,8:52:44 AM
pse,AUDUVPR0,22760,22846,0,49,0,4/16/2007,8:52:38 AM,4/16/2007,8:52:44 AM
pse,USERMNT3,22760,22760,0,120,0,4/16/2007,8:48:59 AM,4/16/2007,8:52:44 AM
pse,CSENTRY ,22760,22760,0,240,0,4/16/2007,8:53:19 AM,4/16/2007,8:53:52 AM
pse,REPPASS ,22760,22869,0,210,0,4/16/2007,8:53:59 AM,4/16/2007,8:54:01 AM
pse,AUDUVPR0,22760,22863,0,49,0,4/16/2007,8:53:55 AM,4/16/2007,8:54:01 AM
pse,CSENTRY ,22760,22760,0,250,0,4/16/2007,8:53:19 AM,4/16/2007,8:54:01 AM
pse,DBMAPMNT,22760,22760,0,19,0,4/16/2007,8:54:29 AM,4/16/2007,8:54:45 AM
```

- ◆ User ID
- ◆ Program Name
- ◆ GID
- ◆ PID
- ◆ Memory Usage
- ◆ CPU Time
- ◆ CPU Usage (always zero)
- ◆ Start Date
- ◆ Start Time
- ◆ End Date
- ◆ End Time


Sample output of a data file exported to XML

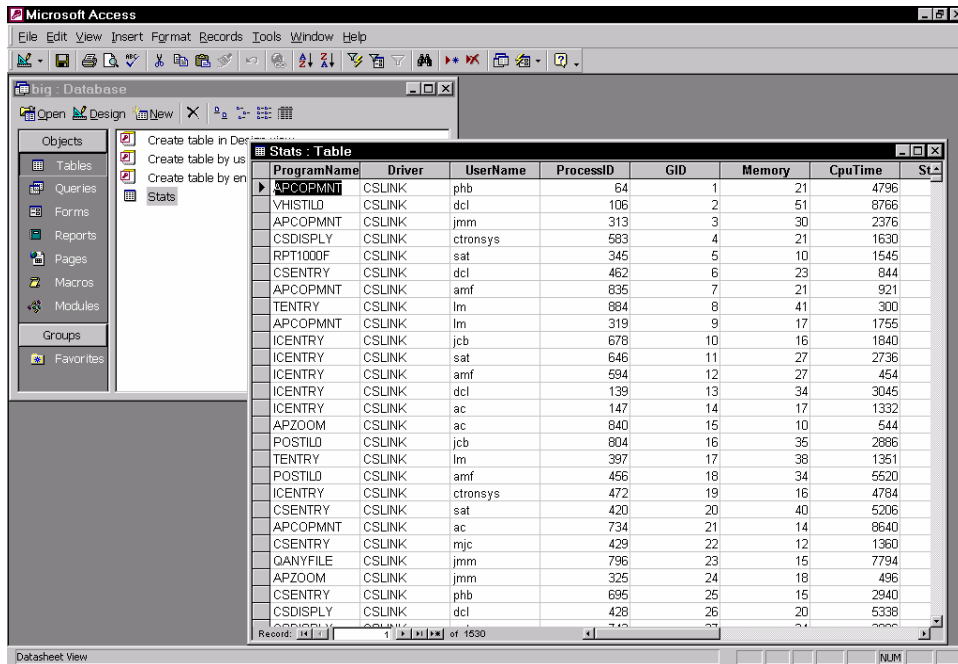
```
<?xml version="1.0" encoding="UTF-8"?>
<Elements>
<Element id="30" User="pse" GID="22760" PID="22760">
  <Program>FORMINT</Program>
  <Driver>CSHSDRVR</Driver>
  <Memory>0</Memory>
  <CpuTime>29</CpuTime>
  <StartDate>4/16/2007</StartDate>
  <StartTime>8:48:16 AM</StartTime>
  <EndDate>4/16/2007</EndDate>
  <EndTime>8:48:16 AM</EndTime>
</Element>
<Element id="31" User="pse" GID="22760" PID="22783">
  <Program>REPPASS </Program>
  <Driver>CSLINK</Driver>
  <Memory>0</Memory>
  <CpuTime>259</CpuTime>
  <StartDate>4/16/2007</StartDate>
  <StartTime>8:48:55 AM</StartTime>
  <EndDate>4/16/2007</EndDate>
  <EndTime>8:48:57 AM</EndTime>
</Element>
<Element id="32" User="pse" GID="22760" PID="22777">
  <Program>AUDUVPR0</Program>
  <Driver>WRUN</Driver>
  <Memory>0</Memory>
  <CpuTime>59</CpuTime>
  <StartDate>4/16/2007</StartDate>
  <StartTime>8:48:49 AM</StartTime>
  <EndDate>4/16/2007</EndDate>
  <EndTime>8:48:57 AM</EndTime>
</Element>
```

 Note: The XML tags identify the fields from the data file.

Using Microsoft Access

- ◆ The Analyzer creates valid Microsoft Access database file (.mdb).
- ◆ It is possible to open this file using Microsoft Access.
- ◆ After your database file is loaded into Access, the data is in a table called “Stats”.
- ◆ This data can then be analyzed using whatever functions Access provides.

 **Warning!** Computron does not provide support or licenses for Microsoft Access. Use of Microsoft Access is at the user's own risk.



Accessing a Analyzer database in Microsoft Access

The CSMasterStat.Dat file layout

The CSMasterStat.Dat file is laid out with pipe delimiters. Here is an example data line:

```
1|ctronsys|459|367|POSTILD|CSLINK|11-15-2006|18:31:18|11-15-2006|18:33:23|170|10|0
```

The fields included here are (in order):

Value	Description
1	version (Always "1" - reserved for future expansion of this structure.)
ctronsys	username (CTRON_LOGNAME)
459	GID (WISPGID)
367	PID
POSTILD	program which was executed
CSLINK	"driver" (the program that started the "program")
11-15-2006	start date
18:31:38	start time
11-15-2006	end date
18:33:23	end time
170	CPU time (millisec)

Value	Description
10	memory usage when program completed (out of 100%)
0	CPU usage when program completed (out of 100%)

☞ *Note:* We are currently unable to get this data from Windows.

Troubleshooting

On some systems, especially those without a recent Microsoft Office installation or Microsoft Internet Explorer installation, it may be necessary to install the latest version of MDAC (Microsoft Data Access Components) and the latest Microsoft Chart Control (MSCHRT20.OCX).

⚠ *Warning!* These components should only be installed if the latest MDAC (i.e. for SQL server) has not been previously installed or if there are problems with the using the Analyzer.

Installing MDAC and MSCHRT20

Typically, these installation files are available on line at www.microsoft.com.

Installing MDAC

Problem: Unable to import records into a database.

Solution: Install latest MDAC. (Some error messages will specifically state that this is necessary.)

☞ *Note:* Version 2.1 of MDAC is the minimum required version for the Analyzer.

Installing MSCHRT20.OCX

Problem: Charts do not display; all you see is a gray screen in chart mode.

Solution: Install MSCHRT20.OCX

☞ *Note:* To install the Microsoft Chart Control, follow these steps:

1. Copy the MSCHRT20.OCX file from the ...\\misc directory to the Windows system directory (typically C:\\winnt\\system32).
2. Register the file by running "regsvr32 mschrt20.ocx".
3. From Windows Explorer, double-click on the MSCHRT20.REG file in the ...\\misc directory.
4. The new Chart Control should now be installed. Re-start the Analyzer to properly utilize it.

5.4 Computron's Configuration (.config) File

Introduction

The behavior of Computron applications software is influenced by several factors: the Computron runtime, application global environment parameters, Unix environment variables and Computron's configuration file. Many of these factors are user-defined or specific to a particular environment, while Computron controls some factors. More specifically:

- ◆ The Computron runtime is controlled by the software releases installed on your system.
- ◆ Unix environment variables are dictated by your Unix environment.
- ◆ Application global environment parameters are user-defined to reflect the required processing environment for each product. (Refer to the appropriate Computron product Operations Guide for more details regarding global parameters.)
- ◆ The Computron configuration file is both user-defined to set preferences of necessary parameters for the environment as well as dictated by the Computron runtime to support various types of implementations.

Computron's configuration file, `/.../exec/_ctron_/config` (where `/...` represents the directory name of the file system where the Computron software resides), globally sets parameters for Computron users when they initiate a Computron session in the given file system. This file stores variables that are likely to change from site to site. Each configuration variable is described in this section.

☞ *Note:* Ideally, the Computron configuration file contains only those parameters that are set to values other than the default. The application software uses the default value for all configuration parameters not explicitly set via the configuration file.

Configuration Variables

The configuration variables are contained in the standard Computron configuration file. The purpose of each variable, its valid values and the default setting of each parameter are described below.

VIDEO_MODE

This variable determines the character screen handler that is used for the Computron session. Computron supports two types of screen handlers: WISP and CURSES. This parameter is used in conjunction with various Unix environment variables as well as Computron files that enable keyboard and display character set mapping to the international (8 bit ASCII) character set. For detailed information regarding the screen handlers, contact your Computron Client Support Representative. Valid values are:

WISP – Use the Computron screen input/output drivers that employ WISP screen handling routines.

CURSES – Use the Computron screen input/output drivers that employ native Unix CURSES routines.

Default: WISP

COMMAND_KEY_PGM

This variable is used when the VIDEO_MODE is set to CURSES and determines the action performed when the command key (Ctrl+E) is executed in a character cell session. Valid options are:

- WUSAGE – This presents the “classic” command menu.
- SHELL – This presents the shell command line.
- NONE – This disallows the use of the command key (Ctrl+E).
- program-name – This executes the named program.

Default: –SHELL

CTRONSYSLOG

This variable names the file that is created to store system error messages. Computron designates certain error messages as system error messages. When these errors occur, the error message is written to the log file of the logged in Computron user (.../var/user_XXX, where XXX represents the Computron user ID) as well as the system log file specified by system_log. Enter the absolute pathname to the file that will be used as the Computron system log file.

Default: /.../\${CTRON_LOGS}/system_log (where /... represents the directory name of the file system where the Computron software resides.

DB_PACKET_SIZE

This variable specifies the network packet size that the Computron designated RDBMS server requests from the RDBMS database. Larger packet sizes improve performance at the expense of memory. A system must be evaluated to determine the memory available for the Computron applications. A value of 4096 enables optimal performance of the Computron applications without excessive use of memory. This variable may be set to any multiple of 512.

Default: 4096

☞ *Note:* Computron's application software currently supports variable network packet sizes for Sybase and MS SQL RDBMS implementations only.

DB_MAX_CONNECTIONS

This variable determines the maximum number of connections to the RDBMS database per Computron user. One connection to the RDBMS database for each Computron application installed on the system is required to run the Computron software; however, performance improves with every additional connection to the database established. A value of 10 enables optimal performance of the Computron applications. This variable may be set to any positive integer.

Selecting a value for the number of connections to the RDBMS database is primarily dependent on the memory available on the Unix system. A memory-constrained system may only be able to support one connection per Computron application. Each additional

connection used by the system requires more memory and can deplete memory used for data caching. The system must be evaluated to determine the amount of available memory.

Default: 1

Notes:

- ◆ Computron's application software currently supports multiple connections to the database for Sybase and MS SQL RDBMS implementations only.
- ◆ Oracle implementations running versions 7.3 or higher have the option of setting the maximum number of database connections to either 1 or 2. A setting of 2 allows the runtime to issue a second database connection for certain database maintenance procedures (e.g., truncating tables or rebuilding key paths).

DB_FORCE_DB_NAMES_LOWER_CASE

This variable determines whether the database name and server name in Computron's database map file is mapped to lower case or used literally from the database map file unless redefined in the LGMAP file. If redefined, the LGMAP file entry is used literally. Valid values are:

YES – Maps database and server names to all lower case regardless of whether they are redefined in the LGMAP file.

NO – Uses database map entries for database name and server name literally, unless redefined in the LGMAP file. If redefined, LGMAP file entry is used literally.

Default: Yes.

FOREIGN_CHAR

This parameter determines whether screen images use the standard international (8-bit ASCII) character set or a foreign character set. If a foreign character set is used, appropriate localized screen images must be present and this variable must indicate that a foreign character set is in use. Valid values are:

Y (Yes) – Use foreign character set.

N (No) – Do not use foreign character set.

L (LowFacs) – Use double-byte character set.

Default: No.

BACK_END_NICE

This parameter specifies the Unix 'nice' value for back-end processes, which determines how much priority to give each application process based on a scale of 1–10.

Default: 5

DB_ENABLE_DYNAMIC_UPDATES

This parameter determines whether to enable dynamic updates based on fields that change. Valid values are:

YES – Use dynamic updates.

NO – Do not use dynamic updates.

Default: No.

CTRON_MAX_OPEN_FILES

This parameter specifies a soft limit on the amount of program files that can be open simultaneously. The amount of open files indicated depends, of course, on your system resource limitations.

Default: 400

CODE_PAGE

This parameter determines whether foreign-character code page is being used by the system. A code page is required for double-byte text for foreign languages with large character sets (e.g., Japanese). If a foreign character code page is used, this variable must indicate which code page set is in use. Some valid values are:

932 – Japanese

936, 950, 949 – For future use.

Default: 0 (English)

CSPROC_ENABLE

This parameter determines whether to enable process logging and PROCLIST file generation. Process logging provides performance analysis for application processes (e.g., which file was opened by whom and when), whenever a program is opened and continually writes this information to the PROCLIST file in the *worklib* directory. Valid values are:

YES – Enable processing logging.

NO – Disable processing logging.

Default: Yes.

CSADMIN_ENABLE

This parameter determines whether to enable CSADMIN and CSUEADMIN UserExit calls.

YES – Enable CSADMIN and CSUEADMIN UserExit calls.

NO – Disable CSADMIN and CSUEADMIN UserExit calls.

Default: NO

CSGLOBV_PROGRAMS

This variable enables the Computron Global Variable server for specific programs or all programs. Valid options are:

program names – list the programs that are to use the Computron Global Variable server.

*ALL – Use the Computron Global Variable server for all programs.

Default: *ALL

WISP_GID_CREATE

This variable generates the WISPGID parameter using either the process ID (PID) or the sequence number (SEQ). Valid options are:

PID – Generate the WISPGID using the process ID.

SEQ – Generate the WISPGID using the sequence number.

Default: PID

ENV

This variable sets environment parameters for the start script. Place the environment variables after the ENV= argument.

CSRDFDR_MAX_BINARY

This variable sets the maximum number of binary characters allowed for the first 2K of a file in a record size scan.

Default: 204

SETLOCALE

This variable sets the local category values. Currently only LC_CTYPE is supported with an empty value string.

DB_LOGIN_TIMEOUT

This variable defines the number of seconds to wait for a login request to complete before returning to the application.

Default: 120.

CSCOMM_TCP_NODELAY

This variable turns on TCP_NODELAY setsockopt option for cscomm reads and writes.

Default: YES.

COLLECT_STATS

This variable turns on statistics collection used by the Analyzer. Statistics are written to the CollectStats.Dat file in the ../var/Stat directory. Valid options are:

YES – Collect statistics for the Analyzer.

No – Do not collect statistics.

Default: NO.

CSPSINFO_INTR

This variable sets the process runtime dynamic requests, i.e., enabled bits.

Default: 255.

CSBATCH_CMD

This variable determines the queue management system for background jobs. Valid values are:

cseq – This is the current queue management system preference.

lp – This is the previous queue management system preference.

Default: lp.

Default Configuration File

Computron provides a utility (`gen_config_file`) that allows you to view the configuration file variables and their current default values. This utility resides in the directory, `/.../exec/_ctron_` (where `/...` represents the directory where the Computron software resides).

Syntax:

```
gen_config_file | more
```

The configuration file below is the default configuration file created by the `gen_config_file` utility as of the publish date of this guide.

Note: Keep in mind that the Computron configuration file contains only those parameters that are set to values other than the default.

```
#####  
#  
#           MODIFY THIS FILE WITH EXTREME CAUTION           #  
#           _____           #  
#  
# THIS FILE SHOULD ONLY CONTAIN EXCEPTIONS TO THE DEFAULT CONFIGURATION. #  
#  
# IF YOU ADD AN ENTRY TO THIS FILE, MAKE SURE TO USE THE CORRECT SYNTAX #  
# AND SPELLING, TO VERIFY THE SYNTAX AND SPELLING, RUN THE PROGRAM #  
# gen_config_file TO GENERATE A FILE CONTAINING THE DEFAULT VALUES. #  
#  
# IF THIS IS AN EXISTING INSTALLATION, DIRECT THE OUTPUT OF #  
# gen_config_file TO /tmp/.config BY USING THE FOLLOWING COMMAND: #  
#   gen_config_file > /tmp/.config #  
#  
# EDIT THE CREATED FILE AND COPY THE REQUIRED ENTRIES INTO THE EXISTING #  
# .config FILE LOCATED ON $CTRON_EXEC/_ctron_. BE SURE TO OMIT THE #  
# COMMENT MARKS ("##") FROM THE ACTUAL VARIABLES WHEN YOU ADD THEM. #  
#  
# IF THIS IS A NEW INSTALLATION, DIRECT THE OUTPUT OF gen_config_file TO #  
# CREATE THE STANDARD .config FILE IN THE DIRECTORY $CTRON_EXEC/_ctron_ #  
# BY USING THE FOLLOWING COMMAND: #  
#   gen_config_file > $CTRON_EXEC/_ctron_/.config #  
#  
# EDIT THE CREATED FILE AND REMOVE ANY UNNECESSARY ENTRIES. VERIFY THE #  
# VALUES OF THE REQUIRED VARIABLES AND REMOVE THE COMMENT MARKS ("##") #  
# FROM THE ACTUAL VARIABLE LINES. #  
#  
# IT IS IMPORTANT THAT NO EXTRANEIOUS ENTRIES BE ADDED OR LEFT IN THE #  
# .config FILE. FOR EXAMPLE, IF AT ONE POINT THE VALUE OF #  
# DB_ENABLE_DYNAMIC_UPDATES WAS SET TO "NO", AND THIS ENTRY WAS #  
# COPIED OR LEFT IN THE FILE, THE PERFORMANCE BENEFITS OF #  
# USING DYNAMIC UPDATES WOULD NOT BE REALIZED. #  
#  
#####  
  
# Video Display Mode: WISP = use WISP v Wang, CURSES = use unix curses  
# VIDEO_MODE=CURSES  
  
# Command Key (Ctrl-E) action for VIDEOMODE=CURSES: -WUSAGE, -SHELL,  
# -NONE, or program-name
```



```
# COMMAND_KEY_PGM=-SHELL

# Computron System Log File
# CTRONSYSLOG=${CTRON_LOGS}/system_log

# Network Packet Size for Communication with Database Server
# DB_PACKET_SIZE=4096

# Maximum Number of Database Connections
# DB_MAX_CONNECTIONS=1

# Allow/Disallow Mixed Case Database Names from Database Map
# DB_FORCE_DB_NAMES_LOWER_CASE=YES

# Set Display of Foreign Characters (Y=Yes, N=No, L=LowFacs)
# FOREIGN_CHAR=N

# Sets Nice Value for Back End Processes
# BACK_END_NICE=5

# Enable/Disable Dynamic Updates Based on Fields that Change
# DB_ENABLE_DYNAMIC_UPDATES=YES

# Soft limit on max number of files to open (HP-UX,SOLARIS,NCR,PYRAMID)
# CTRON_MAX_OPEN_FILES=400

# Foreign Character Code Page (932,936,950,949 ==> Double byte)
# CODE_PAGE=0

# Enable Process logging and proclist file generation
# CSPROC_ENABLE=YES

# Enable csadmin and csueadmin user exit calls
# CSADMIN_ENABLE=NO

# Enable CSGLOBV for specific programs or *ALL
# CSGLOBV_PROGRAMS=*ALL

# Generate WISPGID parameter using process id (PID) or sequence number
# (SEQ)
# WISP_GID_CREATE=PID

# Environment Parameters are placed here for setting in the start script
# ENV=

# Max number of binary chars allowed first 2K of a file in recordsize scan
# CSRDFDR_MAX_BINARY=204

# SETLOCAL Category=value,Category2=value2,... (see man setlocale).
# Currently only LC_CTYPE should be used, and preferably empty
# SETLOCALE="LC_CTYPE="

# Defines the number of seconds to wait for a login request to complete
# before returning to the application
# DB_LOGIN_TIMEOUT=120

# Turns on TCP_NODELAY setsockopt option for cscomm reads/writes
# CSCOMM_TCP_NODELAY=YES

# Turns on statistics collection, writes to file CSMasterStat.Dat
```

COLLECT_STATS=YES

Process Runtime Dynamic Requests (enabled bits).
CSPSINFO_INTR=255

Queue Management System for Background Jobs. Values are: "cseq" and
"lp"
CSBATCH_CMD=cseq
CSBATCH_CMD=cseq

#DB_ENABLE_CONCATENATED_SELECTS=NO
BULKLOAD_ENABLE=YES

5.5 Creating Scripts via Computron's Procedure Generator

Introduction to Scripts

In Computron's application systems, all processing is initiated from a window. That is, when any function is selected from a menu, at least one window is displayed so that you can either proceed with the function or exit without processing. You may prefer, however, to *proceduralize* one or more windows in order to control how a function is run.

For example, if the General Ledger's application Post and Report (TRANSPP0) function is generally run at night by personnel who may not have an understanding of how the TRANSPP0 function works, you can create a procedure that prevents any of the windows from being displayed to the operator. A procedure would pass default values for required fields on the windows to the program and supply action key (F1–F32 or Enter) entries for processing.

☞ *Note:* This sample procedure could also be accomplished by using Computron's Job Processing module. For more details, see the "Procedurally Invoking the Job Processor" instructions later in this section.

You can, of course, create your own procedures using a scripting language. However, Computron's Soft Technology component features a process called the *Procedure Generator*, which provides the capability to generate a shell procedure for application functions. This section steps you through Computron's Procedure Generation process. Keep in mind that a procedure may require some additional editing after it has been created. Therefore, elements that must be manually entered are described in the "Additional Editing Requirements" instructions in this section (e.g., complex commands that cannot be handled by the Procedure Generator).

Structure

To understand how the Procedure Generator is constructed, it is important to know how a procedure interacts with the Screen Driver, Computron's Soft Technology program. Typically, in an interactive mode, the Screen Driver does the following:

1. Displays a window.
2. Waits for a user to enter valid data on the window and press an action key.
3. Validates the entries based on Field Validation (FV) record entries and/or program-imposed requirements.
4. Displays error messages (if necessary) and, optionally, repaints the window to position the cursor at the field(s) in error. This step is performed when validations have failed or the action key causes the same window to display (e.g., F1 to validate and redisplay a window).
5. Proceeds to the next step in the program (e.g., display the next window, update a file or print a report).

When a procedure is used to run a program, it is possible that additional steps may have to be performed. However, for a window to issue a GETPARM (to go and read the procedure for information), one of the following conditions must be true:

1. If the window's Display and Read Screen (DR) record has the Screen I/O Type field set to either *G* (GETPARM) or *H* (Hidden GETPARM), a GETPARM is performed. The GETPARM searches the procedure and attempts to find default values to load into fields on the current window. If there are 'cspuparm DISPLAY' or 'cspuparm ENTER' statements included in the procedure for a window, the values are loaded. Whether the window is displayed, however, depends on the rules described below. If neither type of statement exists in a procedure and the *G* option is selected, a window is displayed with the default values. If neither type of statement exists in a procedure and the *H* option is selected, a window is not displayed unless an error occurs (the window is required to display error messages).
 - ◆ If a procedure includes a 'cspuparm ENTER' statement for a window, the system uses the GETPARM values for the window along with an action key (a function key or, if no function key is specified, the Enter key is assumed). The window is not displayed.
 - ◆ If a procedure includes a 'cspuparm DISPLAY' statement, the window is populated with the GETPARM values extracted from the procedure and the window is displayed. Displaying the window allows you to make additional runtime entries before proceeding to the next step/window.
2. Another condition under which a window issues GETPARM, occurs when the following statement is included in a procedure:

```
enter cshpset option = o
```

or

```
enter cshpset option=b
```

The *O* option searches a procedure for a 'cspuparm ENTER' statement that has a window ID containing an ampersand (e.g., &ACO). Immediately following the ENTER statement are any overrides to values entered via the DS11 window in the window's DR record. For example, the type entered in the **Screen I/O Type** field on the DS11 window can be overridden.

The *B* option operates in the same manner as the *O* option, but it also issues a GETPARM to look for default values for the data fields included on the window itself.

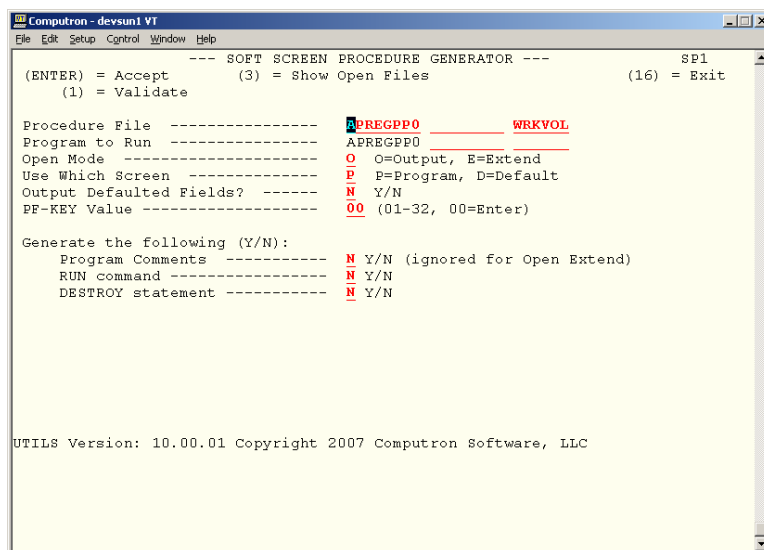
☞ *Note:* Unlike an error message, an information-only message does not cause a window to be displayed. These types of messages can be differentiated from each other in one of two ways:

1. An error-message is accompanied by an audible tone (beep); whereas an information-only message is not.
2. The first two positions in an error message contain a value; the same two positions in an information-only message do not (since they are blank).

Screen Description

The process of creating application procedures is achieved via the Soft Screens Procedure Generator (SP1) window. The Procedure Generator window can only be accessed via Computron's character cell presentation. Follow the steps outlined below to access the Soft Screens Procedure Generator (SP1) window and familiarize yourself with its field values.

1. Start a Computron character cell session.
2. From the menu, run an application function that you would like to proceduralize.
3. Once the initial window is displayed, enter any values that you want written to the procedure.
4. Press PF-31. The current window is displayed in *Set Defaults* mode.
5. Next, press PF-30 to display the Soft Screen Procedure Generator (SP1) window shown below.



```

Computron - devsun1.VT
File Edit Setup Control Window Help
--- SOFT SCREEN PROCEDURE GENERATOR ---          SP1
(ENTER) = Accept      (3) = Show Open Files      (16) = Exit
(1) = Validate

Procedure File ----- APREGPP0      WRKVOL
Program to Run ----- APREGPP0
Open Mode ----- O      O=Output, E=Extend
Use Which Screen ----- P      P=Program, D=Default
Output Defaulted Fields? ----- N      N Y/N
PF-KEY Value ----- 00      (01-32, 00=Enter)

Generate the following (Y/N):
Program Comments ----- N Y/N (ignored for Open Extend)
RUN command ----- N Y/N
DESTROY statement ----- N Y/N

UTILS Version: 10.00.01 Copyright 2007 Computron Software, LLC
    
```

Soft Screen Procedure Generator Window (SP1)

Procedure File

This is the name and location of the file that will be created to store the procedure commands. It is recommended that you store your procedures in a unique library so that they are not overwritten during a subsequent release of Computron software. For example, Accounts Payable scripts could be created in a *cuaprun* directory.

Default: The procedure file is given the same name as the Computron program and is located in a user's default work volume (which is set up via the user's *Usage Constants*). Enter the library location.

Program to Run

This is the name of the Computron program called by a procedure via a RUN Statement. The entry in the library field is used in the SET PROGLIB statement that is generated along with the RUN statement. This field is ignored unless the RUN command field is set to Y (Yes).

Default: The name of the Computron program currently being run. You can enter a specific library and volume name. By leaving them blank, the system uses the standard program library and volume.

Open Mode

This is the mode in which the procedure file is to be opened. Valid values are:

O – Use this option for the first window included in the procedure. The system creates a new procedure as output.

E – Use this option for any window that may be displayed after the first window in a program. Use of this option causes commands to be added to the existing procedure file identified in the Procedure File fields, instead of creating a new file like the O option.

Use Which Screen

This defines the source of the values that are passed by the procedure to the program at run time. Valid values are:

P – Use the values provided by the program as seen on the window when initially displayed.

D – Use the values entered when the window was displayed in Set Defaults (D) mode (when PF-31 was pressed).

Output Defaulted Fields?

This field is used in conjunction with the Use Which Screen field (described earlier), and indicates whether to extract default values from the screen and pass them to the procedure. Valid values are:

Y – Pass defaulted values to the procedure.

N – Do not pass defaulted values to the procedure.

PF-Key Value

This is the number of the action key to be passed to a program by a procedure. The values 01–32 represent function keys. The value 00 represents the Enter key.

Generate the following (Y/N)?

Program Comments

This indicates whether to include comments in a procedure. Valid options are:

Y – Include the Program Comments entry fields at the bottom of the SP1 screen (see the sample screen after Step 6) upon pressing Enter. This option is valid only when the Open Mode field (described earlier) is set to O (i.e., program comments can only be entered when a procedure is initially created).

N – Do not include program comments.

RUN Command

This field is used in conjunction with the Program to Run field (described earlier). Valid options are:

Y – Generate a RUN command. This command tells a procedure to run the program named in the Program to Run field. Use this option when adding the last screen to a procedure.

N – Do not generate a RUN command. Use this option when adding any screen other than the last one to a procedure (i.e., when the Open Mode field is set to E).

DESTROY Statement

This indicates whether to include a DELETE statement in a procedure, which causes the procedure to be automatically deleted upon completion. Valid options are:

Y – Include a DELETE statement at the end of a procedure.

N – Do not include a DELETE statement.

Creating a Procedure

To use the Procedure Generator to create your own procedures, complete steps 1 – 12 as outlined on the following pages. It is important to note that these steps do not apply to windows that are not proceduralized.

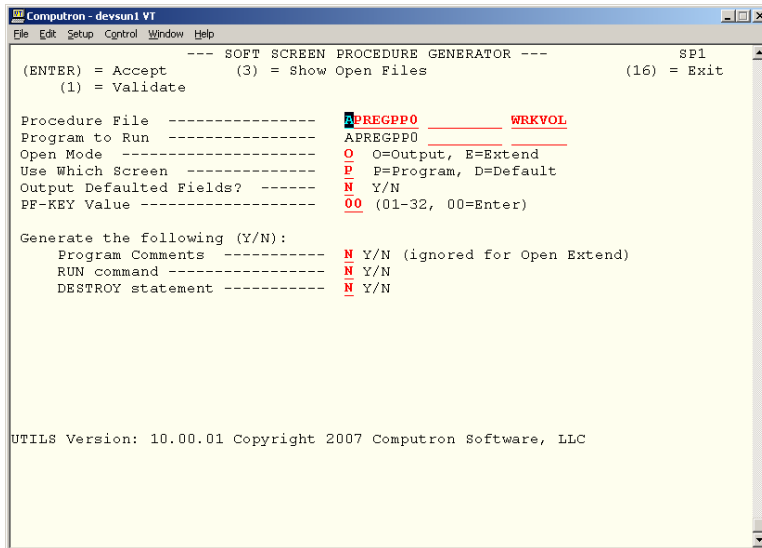
1. Start a Computron character cell session.
2. From its application menu, select the function to be proceduralized. If you know that a front-end program is run from an existing procedure that you would like to include in your customized procedure; i.e. providing values for the GP0 screen, then run the program from Run Any Program to force the GP0 window to display.

When the first window is displayed, decide whether the procedure should display the window. If the window should be displayed, go to step 3; otherwise, go to step 4.

3. Enter data on the current window and press the correct PF-Key to display the next window.

Decide whether the procedure should display the window. If the window should be displayed, repeat step 3; otherwise, go to step 4.

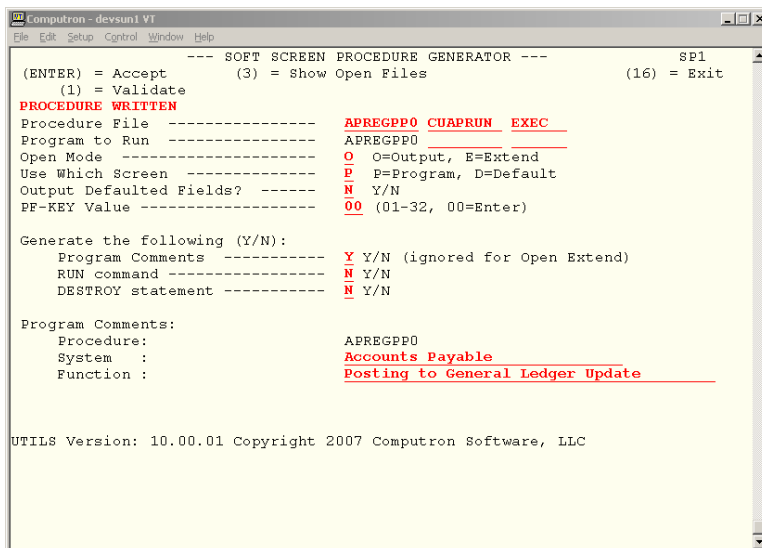
4. Enter data on the window to be written to the procedure.
5. Press PF-31. The current window is displayed in Set Defaults (D) mode. If you enter data on this window, be sure to set the Use Which Screen field to D on the SP1 window.
6. Enter defaults, if any, and press PF-30 to display the Soft Screen Procedure Generator (SP1) window shown here:



Soft Screen Procedure Generator Window (SP1)

7. Using the field descriptions provided earlier in this section as a guide, fill in the pertinent SP1 fields.
8. After filling in the SP1 window (including comments, if used), press Enter to generate a procedure.

Note: If the Program Comments field is set to Y (Yes), the comments fields appear at the bottom of the SP1 window when you press Enter to save a procedure (as shown in the sample below). When these fields are being used, a procedure is not saved until you press Enter again. These field entries are used when generating the Comments area in a procedure and are saved exactly as entered.



Soft Screen Procedure Generator Window (SP1) with Program Comments

9. Press PF-16 twice to exit and return to the original entry window.

10. Repeat steps 3 – 9 for each additional window that you want to include in a procedure.
 - ☞ *Note:* The Open Mode field for all additional windows must be changed from O (output) to E (extended).
 - ☞ *Note:* If the program is to be run after proceduralizing all windows, make sure to set the Run command to “Y” on the last window.
11. After proceduralizing the last window, exit the function and review the generated procedure.
12. Determine what, if any, additional editing is required. For example, editing may be needed to:
 - ◆ Pass additional menu parameters (see following section).
 - ◆ Display a window in a procedure to allow the input of specific fields. (For instance, to replace the display of several Computron windows with a single input window and to supply defaults for all other fields in a procedure.)
 - ◆ Call required back-end programs from a different library (see following section).
 - ◆ Call other programs (e.g., non-Computron utility functions).
 - ◆ Test for conditions.

Additional Editing Requirements

As mentioned earlier in this section, a procedure may require some additional editing after it has been created. Therefore, some elements that must be manually entered are described in this section.

Procedure needs to pass additional menu parameters (e.g., GP0 window parameters).

If the function you are proceduralizing needs to pass additional menu parameters, you are prompted to enter them the first time you attempt to run the procedure in a standalone mode (e.g., Run Any Program). For example, the AP Edit Only Posting (APREGPP0) program needs to know whether it is running in report or update mode, so the procedure presents the GP0 window. (Normally, this window remains hidden when the program is run via the Payment/Posting menu.) To add the GP0 parameters to your script, follow these steps:

1. Run your procedure in standalone mode (e.g., Run Any Program).
2. When the GP0 window appears, enter the appropriate information (e.g., REPORT vs. PROCESS mode).
3. Press PF-31 to switch to Set Defaults (D) mode.
4. Next, press PF-30 to display the Soft Screen Procedure Generator (SP1) window.
5. Append the GP0 window information to the procedure by entering an E (Extend) in the Open Mode field and then press Enter to save it.

6. Exit the Computron function and then review the generated procedure. There should be a line at the end of the script that begins: `csputparm ENTER GP0`
7. Copy and paste the entire line immediately before the `csputparm ENTER CSHPSET` line, as follows:

```
csputparm ENTER GP0 PROCOPT=REPORT OPENMODE=S DATEOUS=GL PFKEY=00  
csputparm ENTER CSHPSET OPTION=G
```

8. Save and exit the script and then rerun it to check for additional errors.

Procedure cannot locate the required back-end program.

When creating custom procedures, Computron recommends that you store them in a unique library so that they are not overwritten by a subsequent release of Computron software. For example, Accounts Payable scripts could be created in a directory called `cuaprun`. This may cause a conflict, however, if the procedure needs to access a back-end program that resides in the standard program directory (e.g., `ciaprun`). If this is the case, you may receive the following message when you run your procedure from the custom directory:

```
FAILED ON LINK TO PROCESS PROGRAM APREGPP0 CUAPRUN EXEC RC=0888
```

To link the back-end program to your procedure, follow these steps:

1. Use your script editing tool to open the procedure and then insert the following line immediately before the `wrun` line:

```
csputparm ENTER GTUSEROV \
```

2. Keep the procedure open in the background while you reopen the Computron function that you are proceduralizing.
3. From any of the function windows, access the WISP Command Processor by pressing `ctrl+e`.
4. From the command prompt, type the following command:

```
csputparm show
```

5. The `csputparm show` command displays program information (`GTUSEROV`) that can be overridden for the current user. Scroll down to locate the following lines:

```
PROGLIB =CIAPRUN  
PASSMSG1=CIAPRUN S AP 000 REPORT N " \
```

These lines tell the procedure to look in the standard program library for the necessary back-end program.

6. Use the copy and paste commands to insert the `PROGLIB` and `PASSMSG1` lines into your procedure immediately after the `csputparm ENTER GTUSEROV` line, as follows:

```

csputparm ENTER GTUSEROV \
PROGLIB =CIAPRUN
PASSMSG1 =CIAPRUN   S   AP   000   REPORT           N   "   \

```

7. Save and exit the procedure and then rerun it to check for additional errors.

Additional GTUSEROV Options

The following GTUSEROV options can also be overridden within procedures:

- ◆ OPFILE
- ◆ OPLIB
- ◆ OPVOL
- ◆ PROG
- ◆ PROGVOL
- ◆ PRTCLASS
- ◆ PRNTMODE
- ◆ PRINTER
- ◆ PSIZEADD
- ◆ IPLVOL
- ◆ WORKVOL
- ◆ WOFFICE

Procedurally Invoking the Job Processor

Computron's Job Processing module consists of several functions that allow you to automate the invocation of successive Computron application processes. Through this module, you can identify any number of Computron application processes to be run in a predetermined sequence, based on specified conditions, without (or with) supervision.

Computron provides data entry windows for initiating the Job Scheduler; however, a system administrator may wish to develop a program that invokes the Job Scheduler without necessitating an operator. Such a program may subsequently be initiated through other means such as Unix cron or Windows scheduler. The following sections describe the steps required to access the Job Scheduler from a Unix script.

The Job Scheduler Directory and Return Code Processing

Refer to Chapter 3, "Job Processing" in this guide to create a Job Processing environment. Create required and optional codes for selecting jobs to run via the Job Scheduler and then enter processes into the directory. See Chapter 3.12 for information on the return code commands.

Create a Procedure

Develop a procedure to pass the desired selection criteria from the Job Processor (KS1) window to the function from within your command file. This procedure becomes the command file that invokes the Job Scheduler.

Edit the Procedure

All Computron application software is launched by a start script. Computron's start script `/.../exec/_ctron_/start`, (where `/.../` represents the file system in which the Computron software resides) invokes Computron server or application software and initializes the environment of the logged in user.

start [-s screen_type] [-i image_type] [-d database_type] [-c custom_prefix] [-f menu_file] [-v(u) port_number] [-m] [-b] [-x] [program_name]

☞ *Note:* For more details regarding the start script, refer to section 5.3, "Computron's Start Script," in this guide.

Edit the procedure you created so that it issues the start command. This is accomplished by activating the commented lines that appear at the beginning of every procedure:

```
# CTRON_EXEC=/ctronxxxxx/exec
# . ${CTRON_EXEC}/_ctron_/start -b -m [other start options]...
  create_worklib
```

Also, you must set the `CTRON_EXEC` parameter to point to the correct Computron directory. Note that the start command already includes the `-m` and the `-b` options. (The `-m` option initializes the operator's Computron environment, and the `-b` option initializes the environment for background processing.)

Sample Procedures

Sample Procedure 1

This section includes a sample procedure that was generated via the Procedure Generator. Editorial comments are interspersed throughout the listing to describe how the various sections of the procedure were generated.

```
#                               G L P O S T 0                               *
#                               *                                           *
# *****                       *                                           *
#                               *                                           *
#      Computron Software, LLC.      *
#      301 Route 17 North,           *
#      Rutherford New Jersey 07070  *
#      (201) 372-6100                *
#                               *                                           *
# *****                       *                                           *
#                               *                                           *
```

The following three comment lines were entered in the fields displayed by setting the Program Comments field to Y (yes).

```
#      Procedure: GLPOST0           *
#      System   : Fixed Assets      *
#      Function  : Script for GL Post *
#                               *
#      Release History:             *
#                               *
#      Level    Date    Rep  Description *
#      -----  -  -  -  -  -  -  -  - *
#      01.01.00 11/16/06  JRS  Initial Release *
#                               *
# *****                       *
#                               *
```

The next series of lines are available if the procedure runs outside of Computron software. They include instructions for activating the comments and directing the procedure to the correct Computron directory.

```
# If this procedure (UNIX shell script) runs outside of Computron Software
# (i.e. standalone), uncomment the next 3 non-blank lines and set
# CTRON_EXEC to the correct Computron directory
#
# CTRON_EXEC=/ctronxxxxx/exec
# . ${CTRON_EXEC}/_ctron_/start -b -m [other start options]...
# create_worklib
```

The ENTER CSHPSET statement is always generated with the Option field set to G, which is the option to do a GETPARM for all windows. Thus, each window called for by the program looks to the procedure for any default values and action keys (Enter key or a function key).

```
csputparm ENTER CSHPSET OPTION=G
```

The next statement was generated for the KS0 window and provides a default value for company number. Because there is no function key entry, the statement assumes that the enter key is used.

```
csputparm ENTER KS0 PFKEY=00 COMPANY="500"
```

The next statement was generated for the KS1, KS2 and KS3 windows, providing default values for each data field on the entry windows.

```
csputparm ENTER KS1 PFKEY=00 FOREBACK="F" SLCTQSW="N" FISPDPP="10" FISPCY="1998"
DATACLAS="ACT" SUMMDETL="S" EDITONLY="Y" BATCHHDR="R" TRANSHDR="R" SOURCEJ="FAJ"
OUTPUTDC=" " BATCHNUM=" " LUPDMODE="P" BATCHRF1=" " BATCHRF2=" " BATCHRF3=" "
GENRPTSW="N" RPTIDVAR="%001" FILENAME="FAGLPOST" FILELIB="CIFAINT" FILEVOL="VOL777"

csputparm ENTER KS2 PFKEY=00 MISCCOD1=" " MISCCOD2=" " MISCCOD3=" " APPLYIND=" "
APPFRTDC=" " APPTODTC=" " MULTSGMT="N" FROMID=" " FROMSS=" " TOID="ZZ ZZ ZZZZ"
TOSS="ZZ ZZZZ ZZZZ ZZZZ" INEXCMSK="I" SSINEXCM="I" IDMASK="** ** *****" SSMASK="**
**** **"

csputparm ENTER KS3 PFKEY=00 INCAASW="Y" INCDESW="Y" INCGLSW="Y" INCITSW="N"
INCIRSW="N" INCXFRSW="Y" INCRRSW="N" INCPLSW="N"
```

The next series of lines represents the wusage values that were manually added to the procedure to link to the back-end program.

```
csputparm ENTER GTUSEROV \
PROGLIB =CIAPRUN
PASSMSG1=CIAPRUN S AP 000 REPORT N " \
```

The next line executes the procedure.

```
wrun glpost0
```

The final series of lines are generated to allow the program to exit from the windows.

```
CTRON_PROG_RETURNCODE=`wretcode`
export CTRON_PROG_RETURNCODE
csputparm clear
```

Sample Procedure 2

This section shows a Job Processor procedure that is run outside of the Computron software environment.

```
# If this procedure (UNIX shell script) runs outside of Computron Software
# (i.e. standalone), uncomment the next 3 non-blank lines and set
# CTRON_EXEC to the correct Computron directory
#
# CTRON_EXEC=/ctronprod/exec
# . ${CTRON_EXEC}/_ctron_/start -b -m [other start options]...
# create_worklib

csputparm ENTER CSHPSET OPTION=G

csputparm ENTER KS1 PFKEY=02 DATEDTC="20061101" PAUSDTC=" " PAUSEHH=" " \
PAUSEMM=" " AMPM=" " INDGRP=" " INDID=" " CLASS=" " CLASS02=" " \
```

```
CLASS03=" " CLASS04=" " CLASS05=" " CLASS06=" " CLASS07=" " \
CLASS08=" " CLASS09=" " CLASS10=" " FREQCODE=" " FREQCO02=" " \
FREQCO03=" " FREQCO04=" " FREQCO05=" " FREQCO06=" " FREQCO07=" " \
FREQCO08=" " FREQCO09=" " FREQCO10=" " APPLICD=" " JOBCAT=" " \
GROUND="D" RUNMODE="D" CALENDAR=" "
```

```
cspuparm ENTER GTUSEROV PROGVOL=EXEC
wrun bsjobpp2
CTRON_PROG_RETURNCODE=`wretcode`
export CTRON_PROG_RETURNCODE
cspuparm clear
delete_worklib
```

5.6 Application Server Directory and File Structure

Introduction to Server Directory and File Structure

The Computron application server requires a very specific directory structure and file placement in order to process successfully. The installation process contains all of the directories and subdirectories in the proper location and hierarchy. The process installs the directories and subdirectories in the required configuration. The top-level directory naming convention has some flexibility, as it is defined in variables that are interpreted in some key configuration scripts and files. But, the directories stored within these key top-level directories should not deviate from the installation structure. As of Release 7.0, significant changes have been made to this structure and this section describes the environment as of this release.

\$CTRON_ROOT

The variable `CTRON_ROOT` is used throughout the Computron server environment to connote the location of the software files and directories. The main file system where the Computron software is installed is exported to the value `CTRON_ROOT` in some key configuration files and is subsequently referred to by the variable `$CTRON_ROOT` (in Unix, the “\$” indicates that it precedes an environmental variable.)

\$CTRON_EXEC or \$CTRON_ROOT/exec

The `exec` directory is located in the file system defined by the `$CTRON_ROOT` variable. This directory is also defined in some key environment configuration files under the variable name `$CTRON_EXEC`. It contains the application specific software (executables) and a single key universal program file called `csuvlpc`.

\$CTRON_ROOT/data

The `data` directory is located in the file system defined by the `$CTRON_ROOT` variable. This directory is not defined by any variables in any of the configuration scripts, but it is defined in the `LGMAP`. Refer to Appendix C, of this guide for more information on the `LGMAP`. It contains pointers to the data files used in the Computron applications.

\$CTRON_ROOT/var

The `var` directory is located in the file system defined by the `$CTRON_ROOT` variable. It was introduced in Release 7.0. It holds several key directories that contain files that do not fall under the executable or data categories. These directories are:

- ◆ `enqspool` – This directory contains directories and files related to the new Background Processing implementation called `cseq`. Refer to Section 5.2 of this guide for more information on `cseq`.

- ◆ logs – This directory contains the user log files. The log file name format is user_{login-id}. Prior to Release 7.0, log files were stored in /tmp and the format was ctron_log_{login-id}
- ◆ spool – This directory contains the user print files. The directories in the spool directory have a name format of {login-id}prt. Prior to Release 7.0, the print files were stored in \$CTRON_EXEC/spool.
- ◆ worklibs – This directory contains the user work libraries, which prior to Release 7.0, were stored in \$CTRON_EXEC/worklibs. The worklibs directory contains sub-directories associated with specific user sessions. The format of the work directory name is “wk#####” (6 digits). The 6 digits are the user’s WISPGID, which is a user specific, session variable assigned to a user’s process when they start the Computron application. If the WISPGID exceeds 6 digits, the first 6 digits are used. If the WISPGID value is less than 6 digits, leading zeroes are used after “wk”.

5.7 Computron's Start Script

Introduction to Computron's Start Script

All Computron application software is launched by Computron's `start` script, which invokes the Computron server or application software and initializes the environment of the logged-in user. The `start` script resides in `/.../exec/_ctron_/start` (where `/...` represents the file system in which the Computron application software resides). The `start` script reads configuration parameters and sets variables appropriately for the application.

A user can initiate the `start` script interactively or this routine can be called from a script (e.g., when the user logs onto the Unix system). When issued to launch the Computron application, the `start` routine also performs the following tasks:

- ◆ Creates a work library for the user.
- ◆ Identifies the log file for the user.
- ◆ Invokes Computron's security subsystem.
- ◆ Reads additional files that are designed to customize `start` for the local processing environment.
- ◆ Starts the application.

The start Command

The actions performed by the `start` routine and some of the variables set by the `start` routine are dependent upon the arguments specified on the command line. The sections below describe each available option.

The syntax of `start` is as follows:

```
start [-s screen_type] [-i image_type] [-d database_type]  
[-c custom_prefix] [-f menu_file] [-u port_number(d)] [-w]  
[-m] [-b] [-r runtime_path] [-x] [-n [+]  
configuration_filename]  
[program_name ]
```

-i

The `-i` option invokes the Computron application software with a runtime that supports open image or laser data technology. Use this option in conjunction with AXS-One's Central and Computron's Workflow products. Cite the imaging technology for this command to initialize as follows:

- ◆ Specify `o`, to indicate Open Image technology. This qualification causes the system to prompt for a network node (the software displays the name of the user's PC as the default).
- ◆ Specify `l`, to designate Laser Data. (*Default.*)

Using the `-i` argument without specifying an image type causes the `start` routine to initiate a runtime that supports laser data technology.

-d

The `-d` option starts a relational database management system (RDBMS) version of the Computron application software. The start routine searches the directory, `/.../exec/_ctron_` (where `/...` represents the directory name of the file system where the Computron software resides), for a file called `.start_database` (where `database` represents the appropriate RDBMS (i.e., oracle, SQL sybase)). This file is an RDBMS-specific shell script that sets RDBMS-specific parameters required by the Computron applications and invokes Computron's database runtime (`/.../exec/_ctron_/wrunora`, `/.../exec/_ctron_/wrunsql`, `/.../exec/_ctron_/wrunsyb` where `/...` represents the file system where the Computron software resides).

For example, type the following command to begin an Oracle version of the application:

```
start -d oracle
```

The start script searches for the Oracle specific shell script, `/.../exec/_ctron_/start_oracle` (where `/...` represents the directory name of the file system where the Computron software resides) unless the file, `/.../exec/_ctron_/wisp/config/wrunconfig`, specifies a different file.

-c

The `-c` option prompts the `start` routine to search a designated `load` directory for customized programs. (`load` directories contain Computron application programs. For example, the programs used by Computron's General Ledger system reside in a directory called `ciglload`.) Users can write custom COBOL programs that interact with Computron's software. In order to install Computron software updates without the risk of overwriting custom programs, custom COBOL programs are stored in custom directories. Computron's convention for naming custom load directories follows the format `ySXXload`, where:

- ◆ `y` indicates that the directory contains custom programs.
- ◆ `s` represents the application system with which the custom programs interact as follows:
 - `p` – Accounts Payable
 - `r` – Accounts Receivable
 - `i` – Direct Invoicing
 - `a` – Fixed Assets
 - `l` – General Ledger
 - `v` – Inventory
 - `o` – Purchasing
 - `b` – Time Billing
 - `t` – Value Added Tax
 - `f` – Workflow
- ◆ `XX` represents two characters reserved for the user's reference.
- ◆ `load` indicates that the files in the directory are programs.

Typically, custom programs for the General Ledger system reside in the directory named `y1XXload`.

☞ *Note:* Specify only the custom prefix (`ySXX`) after the `-c` argument.

The value designated with the `-c` option is added to the value of the ACUCOBOL environment variable `CODE_PREFIX`, thus causing the system to search the `/.../exec/ySXXload` directory, prior to searching the standard Computron `load` directory for standard programs. Additionally, the value of the `CODE_PREFIX` variable is used to locate all subroutines as well as all programs that are called using Computron's `wrun` command (frequently used in scripts and procedures that access Computron software).

When issued without the `-c` argument, the `start` routine reads the file, `/.../exec/_ctron_/.custom` (if it exists) and sets the value of the environment variable `CODE_PREFIX` accordingly.

-f

The `-f` option names the initial menu displayed by the Computron application for this session. Use the Menu File Maintenance function (accessed from the Universal Utilities menu in any Computron financial application) to create and modify Computron menus. Specify an initial menu using the format:

```
start -f [[VOLUME/]library/]filename
```

The `start` routine uses the default menu, `exec/ciuvmenu/start`, when invoked without the `-f` option (i.e., no menu file is designated). If a user specifies an initial menu omitting either the volume or the library, the system assumes that the menu file resides in the library, `ciuvmenu`, on the volume, `exec`.

-u

The `-u` option invokes a daemon that calls the program, `.../exec/_ctron_/csserve`. This program initiates a VBAUI or WEBdesk server process at the supplied port number. For WEBdesk this command is used to start the interface on a web server.

-w

The `-w` option runs the program, `initserve`. A server process submits this option when initiating a connection to the CSSERVE program.

☞ **Important!** Do not use this option interactively. This option is for internal use. Do not use this option unless directed to do so by a Computron representative

-p

The `-p` option runs the program, `initpc`. A server process submits this option when initiating a connection to a client PC.

☞ **Important!** Do not use this option interactively. This option is for internal use. Do not use this option unless directed to do so by a Computron representative

-m

The `-m` option initializes the user's Computron environment, without starting the software or displaying a menu; however, the user's `.profile` file normally runs `start` using this option without user intervention. Frequently, users install Computron software in multiple file systems, thus creating multiple parallel environments. Commonly, new installations set up a testing environment and a live environment. The `-m` option enables users to reset the default environment. In this case, the user can type,

```
. /.../exec/_ctrone_/start [options] -m
```

☞ **Important!** Always precede this command with the character sequence, `.<space>`, in order to invoke the `start` routine and subsequently, initialize your environment within the current shell.

-b

The `-b` option initializes the environment for background processing. (This option is normally issued in conjunction with the `-m` argument.) A typical use of this option is to initialize the environment prior to running procedures without operator interaction (e.g., `cscron`).

-r

The `-r` option initiates a designated runtime, thus overriding the default runtime (e.g., `/.../exec/_ctrone_wruncl2`). Specify the full path of the desired runtime or, if the desired runtime file resides in the directory, `/.../exec/_ctrone_`, specify the filename.

☞ **Important!** This option is for internal use. Do not use this option unless directed to do so by a Computron representative.

-n

The `-n` option causes the `start` routine to read a specified configuration file. Computron's configuration file sets runtime parameters for the given Computron session. Using a plus symbol (+) before the specified configuration filename causes the system to read Computron's default configuration file (`/.../exec/_ctrone_.config`), as well as the specified file. Entering only the specified filename after the `-n` option overrides the default configuration file and reads only the specified file.

☞ **Important!** This option is for internal use only. Do not use this option unless directed to do so by a Computron representative.

program_name

Specifying a program name on the command line causes the `start` routine to launch the application software and display the first window of the program.

Customizing start

☛ **Warning!** *In order to implement new features in Computron applications, Computron reserves the right to overwrite the `start` script; therefore, users must never modify the `start` script itself. Consequently, `start` uses information from several files reserved for customization of the Computron environment.*

Computron's `start` routine reads the following five files. You can modify the functionality of the standard `start` script by editing these files independently or jointly.

```
/.../exec/_ctrone_/local.start
```

and

```
/.../exec/_ctrone_/local.init
```

and

```
./.../exec/_ctron_/.custom
```

(Where `./...` represents the file system where the Computron software resides.)

and

```
<user home directory>/.ctron_start
```

and

```
<user home directory>/.ctron_init
```

(Where `<user home directory>` is the user's login directory set up by the Unix system administrator.)

Since these files are specifically used to customize the `start` routine, they may not exist for all installations. Use a Unix editor (e.g., `vi`) to create any of these files. Note that all files must be readable, and that `local.start` and `local.init` must have the permissions:

```
owner = ctronsys and group = ctron
```

local.start

The `start` routine executes the `local.start` file for all users. Prior to beginning the application software, any commands contained in this file are performed for all Computron users. System Administrator's can exercise control over groups of users or individuals within `local.start` by adding shell programming logic to this file.

local.init

When issued with the `-m` option, the `start` routine executes the `local.init` file for all users. When run with this option, `start` initializes the user's Computron environment, but does not launch the application, display menu or execute the `local.start` file. Use the `local.init` file in place of the `local.start` file when issuing the `start` command with the `-m` option.

When any Computron user invokes `start` with the `-m` option, the `start` routine performs any commands contained in this file. System Administrator's can exercise control over groups of users or individuals within `local.init` by adding shell programming logic to this file.

.custom

In order to prevent overwriting custom software when loading new releases of Computron software, Computron applications access custom software in custom directories. The `start` routine reads the `.custom` file for all users and initializes any custom directories identified within the file. Computron applications access the initialized custom directories prior to the appropriate Computron directories; thus, where custom programs and subroutines exist, they run instead of the standard programs and subroutines.

The `.custom` file contains the four letter custom prefixes used in the installation, one prefix per line. Computron's convention for custom prefixes follows the format: `ySXX`, where:

- ◆ `y` indicates that the directory contains custom programs.

- ◆ S represents the application system for which the directory contains custom programs as follows:
 - p – Accounts Payable
 - r – Accounts Receivable
 - i – direct invoicing
 - a – Fixed Assets
 - l – General Ledger
 - v – Inventory
 - o – Purchasing
 - b – Time Billing
 - t – Value Added Tax
 - f – Workflow
- ◆ xx represents two characters reserved for the user's reference.

The custom prefix values contained within the `.custom` file are added to the value of the ACUCOBOL environment variable, `CODE_PREFIX`, thus causing the system to search the custom directories, `/.../exec/ySX1load` and `/.../exec/ySX2load`, prior to searching the standard Computron load directory for standard programs. Additionally, the value of the `CODE_PREFIX` variable is used to locate all subroutines as well as all programs that are called using Computron's `wrun` command (frequently used in scripts and procedures that access Computron software).

☞ **Important!** When issued with the `-c` option, the `start` routine overrides prefix value(s) defined in the `.custom` file with the prefix value following the `-c` option.

Sample .custom File

The following file contains custom prefixes for Computron's General Ledger, Accounts Payable, and Accounts Receivable systems:

```
ylcs
ypcs
yracs
```

When installed in a file system named, `/ctron`, this file causes the value of the `CODE_PREFIX` variable to begin as follows:

```
CODE_PREFIX=/ctron/exec/ylcsload /ctron/exec/ypcsload
/ctron/exec/yracsload /ctron/exec/_ctron_
```

.ctron_start

The `.ctron_start` file is placed in each user's home directory. The `start` routine reads the `.ctron_start` file after the `.local_start` file. Use this user-specific file to modify the `start` routine or to affect global modifications to the `start` routine resulting from `local_start` for a single user.

.ctron_init

When issued with the `-m` option, the `start` routine reads the `.ctron_init` file after the `.local_init` file. Use this user-specific file to modify the `start` script or to affect global modifications to `start` resulting from `.local_init` for a single user.

5.8 Computron's Cron Job – CSCRON

Introduction to Computron's Cron Job – CSCRON

The `cscron` script runs a supplementary system cleanup operation of the Unix shared memory and temporary files. It resides in the `/.../exec/_ctrn_` directory (where `/...` represents the file system in which the Computron application software resides). This utility is normally loaded into the Unix environment via Computron's installation script, `/.../exec/_ctrn_/ctrn_install`.

Although Computron applications perform a primary cleanup of temporary files, it is strongly recommended that system administrators run `cscron` daily within each Computron software environment. That is, if the Computron software is loaded into a file system mounted at the directory, `/ctrn`, for production and the software is also loaded into a file system mounted at the directory, `/ctrntest`, for testing, then `cscron` must be run separately within each file system (`/ctrn` and `/ctrntest`).

Mechanics

Proper maintenance of the Computron environment requires the system administrator to run `cscron` daily. It is imperative that the system administrator runs this utility when there are no users accessing the Computron software. If any users are logged into the Computron system or if jobs exist in the batch queue, the `cscron` script skips certain cleanup steps.

The program performs the following actions:

- ◆ Checks the system to ensure that cleanup activities will not affect any users currently logged onto the system.
- ◆ Completes a cleanup of all files that the Computron applications create in the `/usr/tmp` and/or `.../var/logs` directory.
- ◆ Deletes any `worklibs` subdirectories that do not belong to an active process.
- ☞ *Note:* The `worklibs` subdirectory is used by the Computron applications as a work area for creating reports. This directory is usually created in the `var` subdirectory of the file system where the Computron software is loaded and is represented as the volume, `WRKVOL`, in the `LGMAP` file.
- ◆ Scans the system for active ACUCOBOL runtimes that are used by Computron applications. If no active runtimes are found, then the system invokes the following utilities:

<code>wsysinit</code>	Performs cleanup activities on the Computron software.
<code>cshldini</code>	Reinitializes the hold list. To protect data integrity, Computron applications automatically set a hold status on certain data when accessed by a user. This utility releases any held resources. <i>Note:</i> The hold list may reside in either shared memory or a disk file.
<code>csclearpp</code>	Performs a cleanup of putparm shared memory segments and related files. Computron applications create labeled blocks of information (called <code>putparms</code>) that are stored in Unix shared memory. Additionally, associated files that locate the shared memory segments are created in the directory, <code>/.../exec/_ctrone_/putparm</code> . Any putparm shared memory segments or related putparm files that exist on the system without an associated active putparm process are cleared by this utility.

- ◆ Truncates the Computron user log files. Computron log files are trimmed to contain only the most recent 2000* lines of the report.
- ◆ The number of lines saved in a user log file can be changed via the log sizes file (`/.../exec/_ctrone_/logsizes`) described under the “Customizing CSCRON” heading of this section.

Customizing CSCRON

The behavior of Computron’s cron job is dependent on several environment parameters and the contents of the log sizes file, `/.../exec/_ctrone_/logsizes`. Since future releases of Computron software may overwrite the `cscron` script in order to implement new features in the software, modifying `cscron` itself is not advised. Alternatively, the following options are available for customizing Computron’s cron job:

- ◆ Use one or more of the files within the software that are reserved for customization of the Computron environment to set the appropriate environmental parameters.
- ◆ Create a custom shell envelope script to set the appropriate environmental parameters and then execute `cscron`. Subsequently, add the name of the custom script to the root crontab entry in order to take advantage of the modifications.

Environment Parameters

Each environment parameter used by `cscron` is described below. Note that these environment parameters are also documented in the `cscron` script shipped with each release of Computron software. Since an updated `cscron` script may be released prior to updated documentation, refer to the `cscron` script on your system for a complete list of environment parameters used by Computron’s cron utility.

WORK_VOL_DIRS

This parameter designates a list of volumes, as defined in the `LGMAP` file (`/.../exec/_ctron_/wisp/config/LGMAP`) that identify user work libraries (e.g., `worklibs`) to be cleaned by `cscron`.

☞ *Note:* As a precaution, `cscron` only performs a cleanup on the specified directories if the native pathname for the volume contains the work sub-string.

Default: `WRKVOL`

CLEANUP_TMPDIR

This parameter determines whether `cscron` performs a cleanup of the directory represented by the environment variable, `$TMPDIR`. Computron applications employ the `WISPSORT` utility for sorting work files. This utility relies on the environment parameter `$TMPDIR` to locate a temporary workspace. The temporary workspace can optionally be cleaned by Computron's cron job. Enter a non-empty string to cause `cscron` to clean the `$TMPDIR` directory.

Default: not set (empty string)

CTRONLOG_DIR

This parameter indicates the name of the directory where the Computron user log files are stored.

Default: `.../var/logs`

CTRONLOG_PREFIX

The fixed part of the Computron log file name to which the Computron user ID of the logged in user is appended. For example, standard installations of Computron software create log files named, `ctron_log_Computron_User_ID`, where `ctron_log` is the value of `CTRONLOG_PREFIX` and `Computron_User_ID` represents the Computron user ID of the logged in user.

Default: `ctron_log`

BATCHLOG_DIR

This parameter indicates the name of the directory where the Computron batch log file (`lpbatch.out`) is stored. The Computron batch log contains information regarding Computron background processing.

Default: `/tmp`

The Log Sizes File (`.logsizes`)

The log sizes file (`/.../exec/_ctron_/ .logsizes`) can be created to specify the size of each Computron log file. Limits can be set for individual user log files, the Computron batch file and the `listprocs` log file. Computron's cron job trims log files that exceed the maximum number of lines to be exactly the maximum number of lines long, removing the oldest information in the log file.

☞ *Note:* The log sizes file is specifically used to customize Computron's cron job and may not exist for all installations. A Unix editor (e.g., `vi`) can be used to create this file. Since this file is not shipped with releases of Computron software, this file will never be overwritten.

The logsizes file is composed of four types of text line entries:

- ◆ Default
- ◆ User
- ◆ Listprocs
- ◆ Batch

Default

The default maximum number of lines that `cscron` keeps in the user log files is represented by the Default entry. The user log files for all users who are not explicitly listed in the log sizes file via user entries (see below) are trimmed to this size by `cscron`. When the log sizes file does not exist or the log sizes file does not contain a Default entry, user log files are trimmed to 4000 lines.

This entry consists of the word `Default` and the number of lines to keep in user log files separated by a space. For example, the following entry causes `cscron` to trim user log files to 1000 lines, unless an entry exists in the log sizes file for a specific user:

```
Default 4000
```

User

The User entry represents the maximum number of lines that `cscron` keeps for a specific user. Creating a user entry for a specific user overrides the Default entry for that user.

This entry consists of the user's Computron user ID and the number of lines to keep in user log files separated by a space. For example, the following entry causes `cscron` to trim the log file for the user, `ctronsys` (CTR), to 2500 lines, regardless of the Default entry:

```
CTR 2500
```

Listprocs

The Computron `listprocs` shell script (`/.../exec/_ctron_/listprocs`) is used to list, and optionally, to kill Unix process IDs and all associated children (descendant) processes. When this script is initiated (either manually or automatically by the Computron applications software), its activity is recorded in the `listprocs` log file (`/tmp/listprocs.log`). The Computron cron job, `cscron`, trims this log file to the size represented by the `listprocs` entry in the log sizes file. When the log sizes file does not exist or the log sizes file does not contain a `listprocs` entry, the `listprocs` log file is trimmed to 1000 lines.

This entry consists of the word `Listprocs` and the number of lines to keep in `listprocs` log file separated by a space. For example, the following entry causes `cscron` to trim the `listprocs` log file to 500 lines:

```
Listprocs 500
```

Batch

The Computron applications software can be configured to record processing information to the batch log file, `lpbatch.out`, when the Background processing option is selected. The Computron cron job, `cscron`, trims the batch log file to the size represented by the Batch entry in the log sizes file. When the log sizes file does not exist or the log sizes file does not contain a Batch entry, the batch log file is trimmed to 1000 lines.

☞ *Note:* The batch log file, `lpbatch.out`, is located by `cscron` based on the value of the environment parameter, `BATCHLOG_DIR`.

This entry consists of the word *Batch* and the number of lines to keep in batch log file separated by a space. For example, the following entry causes `cscron` to trim the batch log file to 3000 lines:

```
Batch 3000
```

Sample Log Sizes File

```
Default 1000  
CTR 2500  
GST 500  
TRN 2000  
Listprocs 500  
Batch 3000
```

Informational Error Messages

`cscron` generates some error messages that can be safely ignored.

- ◆ The following message comes from the start script, which is invoked by `cscron`.

```
Lookup_ctron_id: root is an invalid Computron user.
```

- ◆ The following message appears when the `csldini` utility (to reinitialize hold list entries) is run by `cscron`. It simply indicates that there is no hold list to clean up.

```
Hold list shared memory segment not found (key = ...)!  
Shmget: No such file or directory.
```

Chapter 6

Relational Database

6.1 RDBMS Utilities

Introduction to RDBMS Utilities

In the Computron Relational Database Management Systems (RDBMS) implementations, custom utilities are provided that must be used in the creation and maintenance of the RDBMS Computron application's tables. Using other means to create or alter the Computron application's tables is not advised and **is not** supported. A Computron technical consultant must be involved in any installation or conversion efforts in order for the implementation to be supported.

This section describes various utilities and how they are used. The utilities are listed in a logical order, based on the order in which they would be run to create the Computron application's database. This section describes the following utilities:

- ◆ gen_parmdir
- ◆ dbschema.ini
- ◆ create_tables
- ◆ create_indexes
- ◆ create_db
- ◆ grant_permissions
- ◆ create_role
- ◆ drop_tables
- ◆ truncate_tables
- ◆ drop_sp
- ◆ rdbmap_dbdiff
- ◆ check_indexes
- ◆ table_count

⚠ *Warning: A Computron technical consultant must be involved in any installation or conversion efforts in order for the implementation to be supported.*

Terminology

This document assumes a strong familiarity with some Computron specific concepts and terms. The terms and concepts include:

- ◆ LGMAP
- ◆ Database map
- ◆ User file records { &&A record, Database ID, Database Password }
- ◆ RDBMap
- ◆ Unix shell command line
- ◆ User log file

☞ *Note: If you lack a sufficient understanding and comfort level with these terms, review the appropriate sections of the System Administrator documentation.*

Preparing for the Creation of the Computron Application Tables and Indexes

Application Server Activities

database map

The key to all of the database utilities is the Computron application's database map. The database map specifies the database name, the server name (in certain cases), the table owner and the RDBMap location. Therefore, it is critical that the database map be reviewed and modified prior to the creation of the Computron application's tables and/or indexes. Any custom entries that redirect a file or RDBMap should be made to the database map prior to the creation process.

gen_parmdir

This script creates the parameter directories and the required files are copied/generated onto the Application server. Though most of the data is stored in the relational database (RDB) tables, there are a few files that are still maintained as vision or consecutive files on the application server in directories. These files are usually either sequential (flat) files holding data for a product or temporary work files that get cleared at the end of a process. The `gen_parmdir` script is located in the `$CTRON_EXEC/db_run` directory.

The required parameters for running this script are:

- ◆ Database map
- ◆ Volume entry for source files
- ◆ Product prefix
- ◆ Directory name for source files
- ◆ Full path name of destination directory – this must match the parm file definitions from the database map

If converting from a vision implementation, the source directories are those in which the vision data is located. If creating a brand new RDBMS environment, the demo data contained on the release can be used as the source for the parameter files. If no source data exists (i.e. empty directories are specified), the `gen_parmdir` script creates the required files.

Example:

The source files are located in the directory `/ctron/data/csuvdemo`. The LGMAP entry pointing to `/ctron/data` is `data`. The database map is `ctclprod`. The target location for the parameter files is `csuvprod`. The command to create the parm directories and files for the `csuv` product is:

```
gen_parmdir -v data -d ctclprod csuv csuvdemo /ctron/data/csuvprod
```

The `gen_parmdir` script copies the required files from `csuvdemo` to `csuvprod`. If the required files do not exist in the `csuvdemo` directory, they are created using the command, `vutil -gen`.

&&A record and User File Maintenance

When the database schema and table owner have been determined, a key to this table owner must be specified in the Computron User file. This record is the &&A record and is documented in full in Section 2.1, “User File Maintenance,” earlier in this guide. Refer to the heading, “Special Database User ID Records,” for complete instructions for adding this record.

⚠ *Warning: The &&A record must exist before the tables or indexes can be created or the function returns an error message and the process fails.*

dbschema.ini

Within each RDBMS system, it is common practice to specify the location where tables or indexes should be created. In Oracle, this location is referred to as tablespace, in Sybase, it is segment and in MSSQL, it is filegroup. In addition, in some RDBMS systems, storage clauses can be specified when a table or index is created. In the Computron implementation, the `dbschema.ini` file is the configuration file where these locations and storage clauses are defined. This file is referenced by the Computron database creation utilities.

The `dbschema.ini` file is located in `$CTRON_EXEC/_ctron_`. Even if specific locations and/or storage clauses are not used (the RDBMS system defaults are used), this file is required when using the database creation utilities. In the initial installation of the Computron environment, the `dbschema.ini` file does not exist. The release contains RDBMS specific `dbschema` files (e.g. `dbschema.ora`, `dbschema.syb`, etc). The first time `create_tables` or `create_indexes` is run; the `dbschema` file for that RDBMS system is copied to the file called `dbschema.ini`.

The suggested method to customize the file is to run the `create_tables` or `create_indexes` function in pretend (-P) mode (this is described later in this section). Since the key to the environment does not exist, the user is prompted for the default location (tablespace, segment or filegroup) for the tables or indexes. An RDBMS specific default location is displayed on the window when the key to the environment is being written to the `dbschema.ini` file and the user can accept the default. This method is recommended because the `create_db` program generates both the key to the environment and the location information in the exact format required. This minimizes the possibility of input errors. At that point, a Computron technical consultant, working with the DBA, can make the required customizations to the `dbschema.ini` file.

☞ *Note: The `create_db` program, when run in -P (pretend) mode, does not validate either the location information or the storage clause. The user is responsible for ensuring that the location is valid and the storage clauses are in the correct syntax.*

Unless the information is changed, each time `create_tables` or `create_indexes` is run, the location defined in the `dbschema.ini` file and any storage clause added to the file is used for the creation of the tables/indexes.

In the `dbschema.ini` file, the key to a database environment is:

```
[rdbms={oracle, mssql or sybase} database={database-name}
owner={table-owner-name} server={server-name}]
```

☞ *Note: The parameter `server` is not applicable for all RDBS systems.*

Examples:

1. In an Oracle implementation, with a database instance named `ctcprod` on a dedicated server and a table owner of `ctransys`, with the default locations selected, has the following configuration in the `dbschema.ini` file:

```
[ rdbms=oracle database=ctcprod owner=ctransys ]  
< TABLESPACE ctctables >  
default tables all  
< TABLESPACE ctcindexes >  
default indexes all
```

2. In a Sybase implementation, with a database named `ctcprod`, server called `sybase12_prodsrv` and a table owner of `ctransys`, with the default locations selected, has the following configuration in the `dbschema.ini` file:

```
[ rdbms=sybase database=ctcprod owner=ctransys  
server=sybase12_prodsrv ]  
< on "ctctables" >  
default tables all  
< on "ctcindexes" >  
default indexes all
```

☞ *Note:* The MSSQL configuration has a structure very similar to Sybase.

The `dbschema.ini` file contains documentation that explains in detail the function and structure of this file. Sufficient time should be taken to read and understand the information before customizing the configuration.

Database Creation, Maintenance and Analysis

In order to run the Computron database creation utilities, the processes outlined under the previous heading must be completed. They are:

- ◆ create and verify the database map
- ◆ create the parameter directories and files
- ◆ add the correct `&&A` record for the table owner specified in the database map
- ◆ update and perform required customizations on the `dbschema.ini` file

The basic database creation, maintenance and analysis programs can be run from a function on the System Utilities Menu. The more advanced programs must be run from the shell command line, as they are more powerful and running them improperly or inadvertently can have serious consequences for the data.

Under this heading, all of the programs as run from the shell command line are described. A subsequent heading addresses the programs as run from the menu.

create_db

The `create_db` program is an umbrella executable used to create, maintain and analyze the Computron application's tables and indexes. (It is `create_db.exe` on a Windows platform.) This program includes different programs that perform specific processes. The programs that are available with the `create_db` program can be classified as either creation/maintenance tools or analysis tools.

The creation and maintenance tools consist of:

- ◆ `create_tables`
- ◆ `create_indexes`
- ◆ `grant_permissions`,
- ◆ `drop_tables`
- ◆ `drop_indexes`
- ◆ `drop_sp` (for Sybase and MSSQL systems)
- ◆ `truncate_tables`

The database analysis tools consist of:

- ◆ `check_indexes`
- ◆ `table_count`
- ◆ `rdbmap_dbdiff`

The input parameters for running these programs vary, but the common and required input parameter for all of the programs is the name of the database map.

create_db parameters

The following is a list and description of the parameters for the various `create_db` programs. The parameters that are valid for each particular program are specified in the description of each program.

-m

The `-m` parameter stands for database map and indicates that the value following it is the database map. The database map is required for every one of the `create_db` programs and an error occurs if it is omitted.

-h

The `-h` parameter provides help text. The help is generalized, but provides examples based on the program/script being run.

-r

The `-r` parameter stands for replace and is valid for the `create_tables` and `create_indexes` programs. It performs a drop and create of the objects specified in one step.

-v (1, 2, 3)

The `-v` parameter stands for verbosity, which must be accompanied by a number from 1 to 3. The number indicates that information about the process should be output to the window. The number (level) specified and the program/script being run determine what information is displayed.

- 1 – This level prints general information, such as the full path of the RDBMap(s) being processed, the table being processed, the table owner and other information dependent on the RDBMS system and the program/script being run.
- 2 – This level displays the complete SQL statement that is processed.
- 3 – This level is the same as level 2, but when used with the `-z` option

-P

The `-P` (must be in upper case) parameter indicates that the program is to be run in “pretend” mode. It does a limited the amount of validations, including validating the entries of the database map and validating that the file(s) listed in the command have corresponding RDBMaps (good for catching misspellings). In addition, the `-P` option checks the `dbschema.ini` file to see if a key to the environment exists and if it does not, it prompts for one and creates it. However, in the case of `create_tables` and `create_indexes`, pretend mode does not verify the location (tablespace, segment, filegroup) or the storage clauses specified in the `dbschema.ini`. Also, since it does not actually connect to the database, it is not useful in running any of the maintenance utilities such as `rdbmap_dbdiff` or `check_indexes`.

The `-P` option is commonly used for 2 purposes:

- ◆ To verify the syntax of the command and parameters used
 - ☞ *Note:* It is recommended to use `-P` with the `-v2` parameter to verify any customizations made to the `dbschema.ini` file in terms of syntax. Though it does not perform a validation of the location or storage clause against the RDBMS system, it allows for a visual inspection of the SQL statement.
- ◆ To direct the output of the command (by combining with a verbosity option) to a file for use directly within a SQL tool such as SQLPLUS or ISQL.

-z

The `-z` parameter represents a statement terminator and refers to the terminators used in SQL statements that indicate that the SQL command should be processed (for example, the terminator used in Oracle SQL Plus is the semi-colon symbol, in Sybase it is the word “go”.) This parameter only makes sense when used in conjunction with a verbosity of 2. When used with a verbosity of 2, the method directs the screen output to a file. This file contains an SQL statement that could be run as is from within an SQL session. The `-P` option is also commonly used with the `-v2` and `-z` options to generate the statement but avoid having the SQL statement processed.

- ☞ *Note:* Any of the preceding parameters can be followed by a space and then the parameter’s value or the space can be omitted and the value specified immediately after the option. (e.g. `-m database-map-name` or `-m database-map-name`)

product RDBMap {cixxmap}

The product RDBMap is optional and if not specified, the program is run on the entire database map. The product is specified with the format `cixxmap`, where `cixx` represents the product prefix (e.g. `ciap`, `ciar`, `cigl`, etc) and the suffix (e.g., `map`) represents the RDBMap directory.

The product specification is optional and if no product is specified, the function processes all of the products defined in the database map. The standard practice is to run the programs by

product. Unlike the file lists, only one product can be specified in a single `create_db` command.

logical file or list of files {file file file}

If a product is specified, a logical file or list of files can also be identified. However, the logical files must be related to the product – entering file names that exist in different products are not permitted. The syntax is to specify the product RDBMap followed by a space and then a file name or list of file names all separated by spaces (not commas).

create_db logging

When any of the `create_db` programs are run, a record of the process is written to your log file. This is true for programs run from the command line and when they are run from the Universal Utilities menu. The log can be used to track the activities and results of processing and can be helpful when troubleshooting. The information in the user log includes:

- ◆ Time/date stamp,
- ◆ User ID and process ID of user running the program,
- ◆ Full command entered,
- ◆ Results of the program, including errors.

Creation and Maintenance Programs

The following programs are creation and maintenance programs that come under the `create_db` umbrella program.

create_tables

For each Computron product installed, the corresponding tables must be created via the `create_tables` program. `create_tables` applies Computron's external schema, as defined by the specified RDBMap, to create the database tables. Locations or additional storage clauses specified in the `dbschema.ini` file (described in the previous section) are used in the creation of the table, if they exist.

☞ *Note:* The `create_tables` program must be run before any of the other `create_db` programs, as all of the other programs reference the tables in some manner.

When run from the shell command line, the minimum format for `create_tables` is:

```
create_tables -m {database-map-name}
```

The valid optional parameters for processing the `create_tables` program are:

- ◆ `-v (1,2,3)`
- ◆ `-P`
- ◆ `-z`
- ◆ `cixxmap` - product specification
- ◆ `file, file, file` – A file or a list of individual files can be specified to avoid having the entire product processed.

Examples:

1. The database map is `ctclprod`. The tables being created are for the `ciap` product. The `create_tables` statement is:

```
create_tables -m ctclprod ciapmap
```

2. The database map is `ctclprod`. The tables being created are for the `ciar` product, but only the logical files `aritem`, `applhist` and `corpcust`. You want some general information about the command. The `create_tables` statement is:

```
create_tables -mctclprod -v1 ciarmap aritem applhist corpcust
```

3. The database map is `ctclprod`. The tables are not going to be created, but a SQL statement is generated for testing within an Oracle SQLPLUS session. The product is `cigl` and the table is for the logical file, `ledger`. The SQL statement is written in `/tmp` called `create_ledger_table_script.sql`. The `create_tables` statement is:

```
create_tables -m ctclprod -v 2 -z; -P ciglmap ledger  
>/tmp/create_ledger_table.sql
```

create_indexes

For each Computron product installed, indexes must be created via the `create_indexes` program. The `create_indexes` program applies Computron's external schema, as defined by the specified RDBMap, to create the database indexes. Locations or additional storage clauses specified in the `dbschema.ini` file (described in the previous section) are used in the creation of the index(es), if they exist.

When run from the shell command line, the minimum format for `create_indexes` is:

```
create_indexes -m {database-map-name}
```

The valid optional parameters for processing the `create_indexes` program are:

- ◆ `-v (1,2,3)`
- ◆ `-P`
- ◆ `-z`
- ◆ `cixxmap` - product specification
- ◆ `file, file, file` - A file or a list of individual files can be specified to avoid having the entire product processed.

Examples:

1. The database map is `ctclprod`. The indexes being created are for the `ciap` product. The `create_indexes` statement is:

```
create_index -m ctclprod ciapmap
```

2. The database map is `ctclprod`. The indexes being created are for the `ciar` product, but only the logical files `aritem`, `applhist` and `corpcust`. You want some general information about the command. The `create_indexes` statement is:

```
create_index -mctclprod -v ciarmap aritem applhist corpcust
```

3. The database map is `ctclprod`. The indexes are not going to be created, but a SQL statement is generated for creating the indexes within a Sybase ISQL session. The product is `cigl` and the table is for the logical file, `ledger`. The SQL statement is written to a file in

/tmp called `create_ledger_indexes_script.sql`. The `create_indexes` statement is:

```
create_index -mctclprod -v2 -z go -P ciglmap ledger>
/tmp/create_ledger_indexes.sql.
```

grant_permissions

For each Computron product installed, access privileges must be granted to the database tables using the `grant_permissions` program. The `grant_permissions` program uses the RDBMap to identify the tables and indexes for the specific files or products specified and then generates the appropriate RDBMS command to grant permissions on these tables to the designated user(s).

Like the other `create_db` programs, `grant_permissions` requires the `-m` database map specification to process. It also requires the privilege(s) to be granted and the name of the grantee (usually an RDBMS group, role or database user). The privileges that can be granted on the RDBMS object are: `select`, `update`, `insert`, `delete` or `all` (for all of these privileges).

☞ *Note:* If privileges are specified individually they should be entered in double quotes.

For all privileges to be granted, the word `all` does not require quotes.

Though the `grant_permissions` program supports granting limited privileges, it is not always clear what the impact is within the product.

When run from the shell command line, the minimum format for `grant_permissions` is:

```
grant_permissions -m {database-map-name} {privilege or
list-of-privileges} {user, group, role being granted privilege}
```

The valid optional parameters for processing the `grant_permissions` program are:

- ◆ `-v` (1,2,3)
- ◆ `-P`
- ◆ `-r`
- ◆ `-z`
- ◆ `cixxmap` - product specification
- ◆ `file, file, file` - A file or a list of individual files can be specified to avoid having the entire product processed.

Examples:

1. The database map is `ctclprod`. All privileges are being granted on all products to the users in the `public` group. The `grant_permissions` statement is:

```
grant_permissions -m ctclprod all public
```

2. The database map is `ctclprod`. All privileges are being granted for the `ciar` product for the users in the `ctron` role. General information is to be displayed.

```
grant_permissions -mctclprod -v1 ctron ciarmap
```

create_role

Please refer to the Computron Database Configuration Guide for the Database product you are using.

drop_tables

As new releases of the Computron application software are developed, the need may arise to modify the product tables. There are conversion tools provided for this purpose, but eventually the original table whose structure has been modified needs to be dropped and re-created using a new RDBMap. Another reason for dropping and re-creating tables is to relocate the tables to a different location in the database or with new storage parameters, based on a space or performance analysis. The `drop_tables` program is used for this purpose. Like the other `create_db` programs, it uses the specified RDBMap to identify the database tables associated with a product logical file.

When run from the shell command line, the minimum format for `drop_tables` is:

```
drop_tables -m {database-map-name}
```

⚠ *Warning: The use of `drop_tables` in this manner is not suggested. It is intended for use with limited selection criteria, as this command drops the entire database.*

The valid optional parameters for processing the `drop_tables` program are:

- ◆ `-v (1,2,3)`
- ◆ `-P`
- ◆ `-z`
- ◆ `cixxmap` - product specification
- ◆ `file, file, file` - A file or a list of individual files can be specified to avoid having the entire product processed.

Example:

1. Your database map is `ctclprod`. The tables being dropped are for the `ciar` product, but only the logical files `aritem`, `applhist` and `corpcust`. You want some general information about the command. The `drop_tables` statement is:

```
drop_tables -m ctclprod -v1 ciarmap aritem applhist corpcust
```

drop_indexes

There may be times when indexes for a specific table or group of tables may need to be dropped and recreated. The reasons for this vary, such as troubleshooting a performance issue when an index is in question or relocating the indexes to a different location in the database. Whatever the reason, the `create_db` program provided for this purpose is `drop_indexes`. Like the other `create_db` programs, it uses the specified RDBMap to identify the indexes for the tables associated with a product logical file.

When run from the shell command line, the minimum format for `drop_indexes` is:

```
drop_indexes -m {database-map-name}
```

The valid optional parameters for processing the `drop_indexes` program are:

- ◆ `-v (1,2,3)`
- ◆ `-P`
- ◆ `cixxmap` - product specification

- ◆ *file, file, file* – A file or a list of individual files can be specified to avoid having the entire product processed.

Example:

1. The database map is `ctclprod`. The indexes being dropped are for the tables in the `ciar` product, but only the logical files `aritem`, `applhist` and `corpcust`. The user would like some general information about the command. The `drop_indexes` statement is:

```
drop_indexes -m ctclprod -v1 ciarmap aritem applhist corpcust
```

truncate_tables

The `truncate_tables` program's name is based on the SQL command used in Oracle, Sybase and MSSQL to delete the rows from a table without affecting the structure or configuration of that table. It is not often used in a real production system. It requires the user to have comprehensive understanding of the table(s) being truncated. It is more commonly used in test environments where the integrity of the data is not a consideration. This also requires the user to understand the relationships between data files in order for the data in the test environment to be sensible and produce predictable results. Like the other `create_db` programs, it uses the specified RDBMap to identify the database tables associated with a product logical file.

When run from the shell command line, the minimum format for `drop_tables` is:

```
truncate_tables -m {database-map-name}
```

- ⚠ *Warning: The use of `truncate_tables` in this manner is not suggested. It is intended for use with limited selection criteria or all of the data in all of the tables will be cleared.*

drop_sp (applies to Sybase and MSSQL RDBMS only)

The Computron Sybase and MSSQL runtimes generate stored procedures for the product tables that are re-used by the runtimes for performance purposes. There may be times when the validity of these stored procedures is in question. The user may be advised to run the `drop_sp` command for troubleshooting or performance analysis. Once the stored procedures are dropped, they are recreated dynamically by the runtime when the table(s) is accessed again and a minimum of performance degradation may be noticed before all of the stored procedures are recreated.

When run from the shell command line, the minimum format for `drop_sp` is:

```
drop_sp -m {database-map-name}
```

- ☞ *Note:* In the case of `drop_sp`, specifying only the database map is suggested, since the function is normally used with the intention of removing all stored procedures for the Computron tables.

The valid optional parameters for processing the `drop_sp` program are:

- ◆ `-v (1, 2, 3)`
- ◆ `-P`
- ◆ `cixxmap` - product specification

- ◆ *file, file, file* – A file or a list of individual files can be specified to avoid having the entire product processed.

Example:

1. The database map is `ctclprod`. The stored procedure for the entire database should be dropped. Only errors are required to display. The `drop_sp` statement is:

```
drop_sp -m ctclprod
```

Database Analysis

The following database analysis utilities are not intended to replace the normal monitoring of the RDBMS system that should be performed by the on-site Database Administrator (DBA). They are provided for use by the DBAs, Computron administrators and technical representatives to monitor the environment in specific areas.

rdbmap_dbdiff

It is recommended that the `rdbmap_dbdiff` utility be run on a regular basis as part of normal system maintenance. There may be times when the user or technical representative is advised to run this function as part of analyzing a performance or processing issue.

This function performs a lot of checks of the Computron application environment. These include:

- ◆ Validates that the table structure matches the schema defined in the RDBMap
- ◆ Validates the existence and structure of indexes (this feature may decrease the need for running `check_indexes`)
- ◆ Reports on any external (non-Computron) triggers, properties or additional indexes on a product table
- ◆ Reports on any table that is not represented in the database map when level 1 verbosity is specified

When no verbosity is specified, **only** errors display. This is recommended for greater readability.

When run from the shell command line, the minimum format for `rdbmap_dbdiff` is:

```
rdbmap_dbdiff -m {database-map-name}
```

☞ *Note:* This is the recommended method for running this program. It checks the indexes on every table for every product specified in the database map. Since no verbosity is specified, only the errors display. Since this function also checks the indexes, it can be run in lieu of the `check_indexes` function.

The valid optional parameters for processing the `rdbmap_dbdiff` program are:

- ◆ `-v (1, 2, 3)` (not recommended if only checking for errors)
- ◆ `-P`
- ◆ `cixxmap` - product specification
- ◆ *file, file, file* – A file or a list of individual files can be specified to avoid having the entire product processed.

Example:

1. The database map is `ctclprod`. The integrity of the entire database should be checked. Only errors are required. The `rdbmap_dbdiff` statement is:

```
rdb_dbdiff -m ctclprod
```

check_indexes

It is recommended that the `check_indexes` program be run on a regular basis as part of normal system maintenance. There may other times when the user or technical representative is advised to run this program as part of analyzing a performance or processing issue. Though the `rdbmap_dbdiff` program checks the indexes, it is a more time-consuming process, so there may still be times when only `check_indexes` is required.

When run from the shell command line, the minimum format for `check_indexes` is:

```
check_indexes -m {database-map-name}
```

- ☞ *Note:* This is the recommended method for running this program. It checks the indexes on every table for every product specified in the database map. Since no verbosity is specified, only the errors display.

The valid optional parameters for processing the `check_indexes` program are:

- ◆ `-v (1,2,3)` (not recommended if only checking for errors)
- ◆ `-P`
- ◆ `-z`
- ◆ `cixxmap` - product specification
- ◆ `file, file, file` - A file or a list of individual files can be specified to avoid having the entire product processed.

Example:

1. The database map is `ctclprod`. The indexes for the entire database should be checked. Only errors (missing indexes) are required. The `check_indexes` statement is:

```
check_indexes -m ctclprod
```

table_count

The `table_count` program allows a user to count the rows in a database table or tables. The instances where this would be desired would vary, but common uses are to verify a conversion process or monitor space requirements.

When run from the shell command line, the minimum format for `table_count` is:

```
table_count -m {database-map-name}
```

The valid optional parameters for processing the `check_indexes` program are:

- ◆ `-v (1,2,3)` - In this program, if no verbosity is specified and there are multiple tables for the logical file, only the counts for the individual tables display. For a grand total representing the total of all the records in all of the tables (i.e. in the logical file), a verbosity of 1 is used.
- ◆ `-P`

- ◆ -z
- ◆ *cixxmap* - product specification
- ◆ *file, file, file* – A file or a list of individual files can be specified to avoid having the entire product processed.

Example:

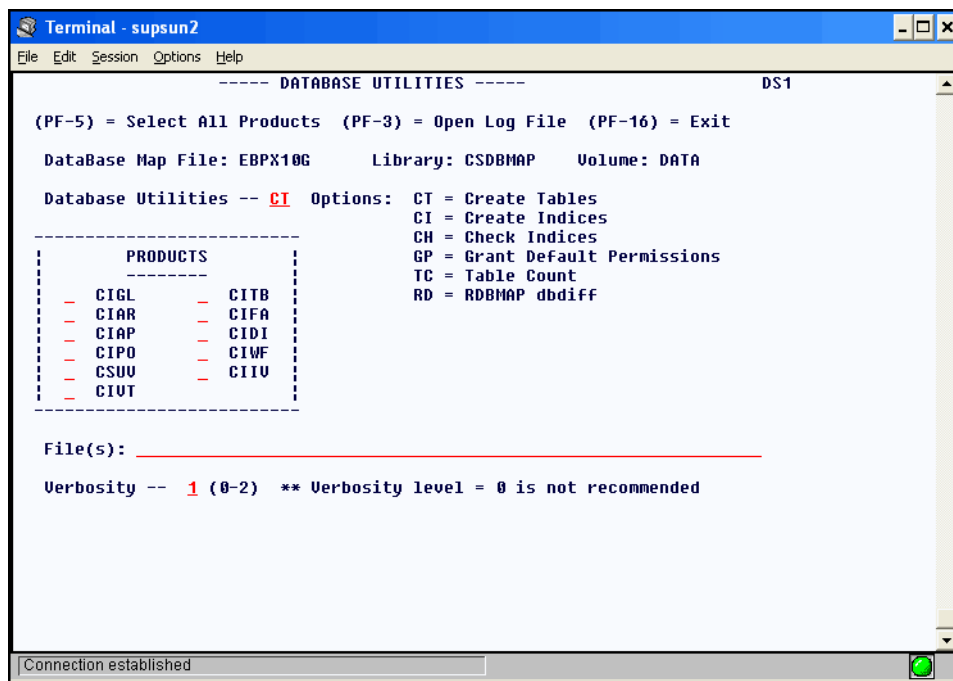
1. The database map is `ctclprod`. The technical consultant has a new RDBMap for the GL file, `trans` and wants to determine how much data is in the table. There are 2 tables specified in the RDBMap for this logical file. A total for both tables is needed. The `table_count` statement is:

```
table_count -m ctclprod -v1 ciglmap trans
```

Database Maintenance on the System Utilities Menu

In addition to the shell command line method of running the database maintenance and analysis tools, most of the basic processes can be run from the Computron Universal Utilities menu. The program can be found in the character cell presentation on the Universal Utilities Menu, Systems Utility Menu, under the title *Database Maintenance*. The Database Utilities on this window behave the same way as those described under the preceding headings, but the input required differs.

Database Utilities Window (DS1)



Database Utilities Window (DS1)

When running the Database Maintenance function from the Universal Utilities menu, the database map is not an input parameter it is based on the database map that the application is set to. To change the database map, PF-29 from a menu and enter the database map. From the Database Utilities window (DS1), you can either process one of the options or view the log file.

This function is useful for troubleshooting. For example, if indexes need to be examined when performance is an issue or the tables need to be compared with the RDBMap because of a product error, any user who has access to this function can run `check_indexes` or `rdbmap_dbdiff` without being conversant with shell command line entry.

Database Utilities

Any option run is run on all of the tables/indexes for the product specified. The options available, listed on the right side of the window are:

- ◆ `create_tables`
- ◆ `create_indexes`
- ◆ `check_indexes`
- ◆ `grant_permissions`
- ◆ `table_count`
- ◆ `rdbmap_dbdiff`

The options not available from this window are:

- ◆ `drop_tables`
- ◆ `drop_indexes`
- ◆ `truncate_tables`
- ◆ `drop_sp`

Products

The products available are listed on the left. One or more products can be specified or all products can be processed at once. If files are listed, only one product can be selected.

☞ *Note:* If the list of products does not show one of the Computron products in your environment (e.g., wfar, wfgl, etc), the product's system prefix can be added to the window via Soft Screens Maintenance.

Files

You can list individual logical files to process. But if files are specified, only one product can be selected. The number of files listed is limited to the length of the input field.

Verbosity

The verbosity level determines that amount and type of information presented. The results of running the option is written to the user's log file, just as when the `create_db` programs are run from the shell command line. Valid values are:

- 0 – Reports on errors only.
- 1 – Lists each RDBMap being processed.
- 2 – Lists the complete SQL statement being run and the full results.

Example:

1. To create the tables in this process, place an X next to the product(s) for which you are creating tables and enter CT in the Database Utilities field. If you wish, you can process all of the products by selecting

PF-05, but this is not recommended. A verbosity level of 1 is recommended. Press Enter (Return) to start the process. When the process is complete, review the log file using the PF-03 command.

6.2 RDBMS Load and Unload Utilities

Introduction to RDBMS Load and Unload Utilities

In the Computron RDBMS implementations, anytime there is a need to populate or retrieve data from any of the Computron application's tables for an initial implementation or as the result of changes to the table structure, the utilities provided for this purpose must be used. Using other means to load, unload, or reload the Computron application's tables is not advised and **is not** supported. A Computron technical consultant must be involved in any installation or conversion efforts in order for the implementation to be supported.

This section describes the various utilities and how they are used. The scripts are listed in a logical order based on the order in which they would be run to create the Computron application's database. This section describes the following utilities:

- ◆ csload_data
- ◆ csunload_data

⚠ *Warning: A Computron technical consultant must be involved in any installation or conversion efforts in order for the implementation to be supported.*

Terminology

This document assumes a strong familiarity with some Computron specific concepts and terms. The terms and concepts include:

- ◆ LGMAP
- ◆ Database map
- ◆ User file records { &&A record, Database ID, Database Password }
- ◆ RDBMap
- ◆ Unix shell command line
- ◆ User log file
- ◆ cconv

☞ *Note: If you lack a sufficient understanding and comfort level with these terms, review the appropriate sections of this *System Administrators Guide*.*

csload_data

There are various situations when the csload_data script is required. Most commonly, this script is used for:

- ◆ Converting from one release of the Computron application to a newer release.
- ◆ Modification of an RDBMap and conversion of the data.
- ◆ Initial installation of an environment.
- ◆ Conversion of a vision environment to RDBMS.

The csload_data script is essentially a wrapper for the COBOL cconv program and calls several programs to generate the required cputparms passed to cconv. It automates and simplifies the

process of populating existing RDBMS table(s). The tables can be populated from files that are in either vision or sequential form. Unlike the `create_db` programs, which are platform and RDBMS specific executables, `csload_data` uses the Computron RDBMS-specific runtime. It can happen that the `create_db` programs work properly, but `csload_data` does not, so the distinction between the utilities (platform specific executable vs. shell script) and the fact that the load data script uses the runtime should be kept in mind.

In order to run the `csload_data` script, the user must exist in the Computron user file and have a security level of 9. The `csload_data` script uses the program, `csuser_info`, to verify these conditions and gives an appropriate error message if either of these conditions is not met (see Section 6.3 of the *System Administrators Guide* for more details on the `csuser_info` program.)

The following is a list and description of the required parameters for the `csload_data` program.

-m

The `-m` parameter identifies the value following it as the database map. The database map is required for this script and an error occurs if it is omitted. The script uses the `csgetaloc` program to interpret the database map from the shell command line (see Section 6.3 of the *System Administrators Guide* for more details on the `csgetaloc` program.)

-h

The `-h` parameter provides help text. The help is generalized, but provides examples based on the program/script being run.

-v

The `-v` parameter represents the logical volume where the input file(s) directory exists. This volume must be a valid file system represented in the LGMAP. The script uses the `wname` function to interpret the LGMAP from the shell command line (see Section 6.3 of the *System Administrators Guide* for details of the `wname` program).

☞ *Note:* The `-v` option in the case of `csload_data` does not relate to verbosity. There is no command for specifying verbosity in `csload_data`.

-l

The `-l` parameter represents the directory where the input file(s) exists. This directory must exist in the file system represented by the `-v` (volume) entry.

-f

The `-f` parameter represents the actual file name of the input file. Typically the input file name is the same as the RDBMap file name, but in some cases the input file name may be different. The `-f` parameter is used with the actual name of the input file and the RDBMap file name is specified after the application RDBMap directory name. The restrictions on the use of `-f` are:

- ◆ This can only be used when loading a single file.
- ◆ The actual file name cannot exceed eight (8) characters.

-c

The `-c` parameter indicates that commit count that is used when loading the data. A higher commit count improves performance, but in the case of Sybase and MSSQL, the size of the transaction log must also be taken into consideration.

-info

The `-info` parameter provides version information about the `csload_data` script.

☞ *Note:* The preceding options can be followed by a space and then the value or the space can be omitted and the value specified immediately after the option (e.g., `-mdbmap` or `-m dbmap`).

Product RDBMap - {cixxmap}

The product RDBMap is not optional; only one product can be loaded at a time. The product is specified with the format `cixxmap`, where `cixx` represents the product prefix (e.g. `ciap`, `ciar`, `cigl`, etc.) and the suffix (`map`) represents the RDBMap directory.

logical file or list of files - {file file file}

A logical file or list of files can also be identified. If a specific file is not identified, the database map is used and the `csload_data` program attempts to find an input file for every logical file defined in the RDBMap directory. The syntax is to specify the product RDBMap followed by a space and then a file name or list of file names all separated by spaces (not commas).

When run from the shell command line, the minimum format for `csload_data` is:

```
csload_data -m database-map-name -v volume -l directory-name -c 1000 cixxmap
```

Examples:

1. The database map is `ctclprod`. The tables being loaded are for the entire `ciap` product. The input files are located in a directory called `ciapdata`, which is located in the file system `/ctron1/data`. The LGMAP entry that points to `/ctron1/data` is `DATA`. The runtime commits the transaction every 5,000 records. The `csload_data` statement is:

```
csload_data -m ctclprod -v data -l ciapdata -c 5000 ciapmap
```

2. The database map is `ctclprod`. The tables being loaded are for the `cigl` product, but only the files `dimref` and `postacct`. The input files are located in a directory called `cigldata`, which is located in the file system `/ctron1/data`. The LGMAP entry that points to `/ctron1/data` is `DATA`. The runtime commits the transaction every 5,000 records. The `csload_data` statement is:

```
csload_data -mctclprod -vdata -lcigldata -c 5000 ciglmap dimref postacct
```

3. The database map is `ctclprod`. The table being loaded is for the `cigl` product. The input file name is actually `ledgd01` and is located in a directory called `/glunload`, which is in the file system `/ctron2/data`. The LGMAP entry that points to `/ctron2/data` is `DATA2`. The runtime commits the transaction every 10,000 records. The `csload_data` statement is:

```
csload_data -mctclprod -v data2 -l ciapdata -f ledgd01 -c 10000 ciglmap ledger
```


csunload_data

There are various situations when the `csunload_data` script is required. Most commonly, this is used for:

- ◆ Conversion from one release of the Computron application to a newer release.
- ◆ Modification of an RDBMap and conversion of the data.
- ◆ Initial installation of an environment.
- ◆ Conversion of a vision environment to RDBMS.

The `csunload_data` script is essentially a wrapper for the COBOL `csconv` program and calls several programs to generate the required `csputparms` passed to `csconv`. It automates and simplifies the process of populating existing RDBMS table(s). The tables can be populated from files that are in either vision or sequential form. Unlike the `create_db` programs, which are platform and RDBMS specific executables, `csunload_data` uses the Computron RDBMS-specific runtime. It can happen that the `create_db` programs work properly, but `csunload_data` does not, so the distinction between the programs/scripts (platform specific executable vs. shell script) and the fact that the `csunload_data` script uses the runtime should be kept in mind.

In order to run the `csunload_data` script, the user must exist in the Computron user file and have a security level of 9. The `csunload_data` script uses the program, `csuser_info` to verify these conditions and gives an appropriate error message if either of these conditions is not met.

The following is a list and description of the required parameters for the `csunload_data` script:

☞ *Note:* These options can be followed by a space and then the value or the space can be omitted and the value specified immediately after the option e.g., `-m dbmap` or `-mdbmap`.

-m

The `-m` parameter identifies the value following it as the database map. The database map is required for this script and an error occurs if it is omitted. The script uses the `csgetaloc` program to interpret the database map from the shell command line (see Section 6.3 of the *System Administrators Guide* for more details on the `csgetaloc` program.)

-h

The `-h` parameter provides help text. The help is generalized, but provides examples based on the program/script being run.

-v

The `-v` parameter represents the logical volume where the output file's directory is located. This volume must be a valid file system location represented in the LGMAP. The script uses the `wname` program to interpret the LGMAP from the shell command line (see Section 6.3 of the *System Administrators Guide* for details of the `wname` program).

☞ *Note:* The `-v` parameter in the case of `csunload_data` does not relate to verbosity. There is no command for specifying verbosity in `csunload_data`.

-l

The `-l` parameter represents the directory where the output file is written. If this directory does not exist, the `csunload_data` script creates it in the file system represented by the `-v` (volume) entry.

-info

The `-info` parameter provides version information about the `csunload_data` script.

Product RDBMap - {cixxmap}

The product RDBMap is not optional and only one product can be unloaded at a time. The product is specified with the format `cixxmap`, where `cixx` represents the application prefix (e.g. `ciap`, `ciar`, `cigl`, etc) and the suffix (e.g., `map`) represents the RDBMap directory.

logical file or list of files - {file file file}

A logical file or list of files can also be identified. If a specific file is not identified, the database map is used and the `csunload_data` script creates and populates a data file for every logical file defined in the RDBMap directory. If the table being unloaded is empty, an empty file is created. The syntax is to specify the product RDBMap followed by a space, and then a file name or a list of file names all separated by spaces (not commas).

When run from the shell command line, the minimum format for `csunload_data` is:

```
csunload_data -m database-map-name -v volume -l directory-name cixxmap
```

Examples:

1. The database map is `ctclprod`. The tables being unloaded are for the entire `ciap` product. The input files are located in a directory called `ciapdata`, which is located in the file system `/ctron1/data`. The LGMAP entry that points to `/ctron1/data` is `DATA`. The `csunload_data` statement is:

```
csunload_data -m ctclprod -v data -l ciapdata ciapmap
```

2. The database map is `ctclprod`. The tables being unloaded are for the `ciar` product, but only the files `aritem` and `applhist`. The output files are to be written to a directory called `ciarunld`, which is created in the file system `/ctron1/data`. The LGMAP entry that points to `/ctron1/data` is `DATA`. The unload is required due to a change in the RDBMap. The new RDBMaps are located in a file system defined in the LGMAP as the logical volume `RELMAP`. The `csunload_data` statement is:

```
csunload_data -mctclprod -vdata -lciarunld -r relmap ciarmap aritem applhist
```

6.3 Miscellaneous Utilities

Introduction to Miscellaneous Utilities

In the Computron RDBMS implementations, various auxiliary utilities were created either for use by some of the primary RDBMS utilities such as `csload_data` or `csunload_data`, or for analysis purposes when upgrading or customizing the Computron software. This section describes the following utilities and how they are used:

- ◆ `print_rdbmap`
- ◆ `rdbmap_diff`
- ◆ `csuser_info`
- ◆ `csgetaloc`
- ◆ `wname`
- ◆ `changerec`

print_rdbmap

The `print_rdbmap` program is a method for printing a description of the RDBMap. The `print_rdbmap` program is located in the directory `$CTRON_EXEC/db_run`. It is common to all RDBMS implementations, but it is platform specific. The only prerequisite for running this program is that the correct database environmental variables are exported using the `../start -d{database}` command. The syntax for `print_rdbmap` is:

```
print_rdbmap {rdbmap-LGMAP-entry} { rdbmap-directory} { rdbmap-file-name}
```

There are some optional parameters for the `print_rdbmap` program, but these are not commonly used. The parameters should be listed before the RDBMap volume, directory and file name.

-a

This parameter prints the DDL names associated with the column name.

-nvp

This parameter prints the information in a string format for parsing.

-r

This parameter prints the information about this RDBMap relative to the release version specified.

-v

This parameter prints the version number of the `print_rdbmap` program.

☞ *Note:* In the case of this program, the `-v` parameter is not used to specify a volume entry from the LGMAP.

Examples:

1. The RDBMap location is specified in the LGMAP as RDBMAP. The product RDBMaps are in the ciglmap directory. The RDBMap is ledger. The user does not want to see the DDL field names. The print_rdbmap statement is:

```
print_rdbmap rdbmap ciglmap ledger
```

2. The RDBMap location is specified in the LGMAP as RDBMAP. The product RDBMaps are in the ciglmap directory. The RDBMap is ledger. The user wants to see the DDL field names and wants to have the file in a format where it can be parsed and manipulated for other comparisons. The print_rdbmap statement is:

```
print_rdbmap -nvp -a rdbmap ciglmap ledger
```

rdbmap_diff

This script does a comparison of two RDBMaps and prints out the differences. The rdbmap_diff script is located in the directory \$CTRON_EXEC/db_run. It is common to all RDBMS implementations, but being a shell script, it is not platform specific. The only requisite for running this utility is that the correct database environmental variables are exported using the .../start -d{database} command.

The rdbmap_diff script calls the print_rdbmap utility to compare two RDBMaps and report on any differences between the maps. The two RDBMaps specified must have the same name and must be contained in the directories with the same name, but located in two separate file systems (volumes). The volumes for the two RDBMaps can be specified using the full path name or the LGMAP entries that point to these volumes. The directory name and the RDBMap name are specified once in the command. The syntax for rdbmap_diff is:

```
rdbmap_diff {vol-1 full-path or LGMAP} {vol-2 full-path or LGMAP} {cixxmap}  
{rdbmap-name}
```

The valid optional parameter for rdbmap_diff is:

-v1

Verbosity – This parameter prints out the details of the differences.

☞ *Note:* If verbosity is not specified, the process just displays the name of the file or files in which differences were found.

Examples:

1. The original RDBMap is located in the standard RDBMAP volume, /ctron/exec/rdbmap/release, which is defined in the LGMAP with the entry RDBMAP. The new RDBMap was copied to /tmp and there is no LGMAP entry that points to /tmp. The RDBMap name is pohdr and it is in the directory cipomap in both cases. A list of the differences, if any, is required. The rdbmap_diff statement is:

```
rdbmap_diff -v1 rdbmap /tmp cipomap pohdr
```

2. An entire product's set of RDBMaps is being compared. The original RDBMaps are located in the standard RDBMAP volume, /ctron/exec/rdbmap/release, which is defined in the LGMAP with the entry RDBMAP. The new RDBMaps are located in the file system (path), /ctron/exec/rdbmap/released, which is

defined in the LGMAP with the entry RELMAP. The application being reviewed is ciarmap. Only the files with differences should be displayed. The rdbmap_diff statement is:

```
rdbmap_diff rdbmap relmap ciarmap
```

csuser_info

The csuser_info program is used in some of the main Computron RDBMS utilities, most importantly the csload_data and csunload_data scripts. The program is located in the \$CTRON_EXEC/_ctron_ directory. It is a method of reading the Computron user file from the shell command line and extracting specific information, which must be specified in the command. The syntax is:

```
csuser_info {KEYWORD}
```

In the syntax above, there is no user specified (which is done with the -u option, described below), so it is assumed the user being queried is the user running the command.

Keyword

The KEYWORD refers to the information that is being read from the User file record. Only one keyword can be specified at a time. The possible keywords are:

DBMAP or ULFIL – Displays the user's default database map.

DBLIB or ULLIB – Displays the library where the user's default database map is located.

DBLIB or ULVOL – Displays the LGMAP volume where the user's default database map is located.

DBUID or ULMDBID – Displays the user's database ID.

SL – Displays the user's security level.

The optional parameters available for csuser_info are:

-u

A specific user can be identified using the 3-character user key. If the -u option is omitted, the current user's login ID is used. This login ID can be determined by typing the word `id` at the shell command line. If multiple user file records are found matching the 3-character user key, the first sorted value is used.

-h

This provides the required syntax and also the information that can be extracted from the user file record in the form of a list of keywords.

☞ *Note:* If the user running csuser_info from the shell command line does not have a security level of 9 or if this user does not exist in the Computron user file, the csuser_info program errors out with the a message to that effect. Since the csload_data and csunload_data scripts call csuser_info, they would also fail for these reasons.

csgetaloc

The csgetaloc program is a method of reading the Computron database map from the shell command line and extracting specific information, which must be specified in the command. It is a program used in

some of the main Computron RDBMS utilities, most importantly the `csload_data` and `csunload_data` scripts. The program is located in the `$CTRON_EXEC/_ctron_` directory. The syntax is:

```
csgetaloc {system-prefix} [filename] {database-map-name} {database-map-
library} {database-map-volume}
```

This program returns the location defined in the database map for the application and, if entered on the command line, the file specified. It can read the database map without running the application and is useful for troubleshooting many of the other utilities such as the `create_db` and `csload_data` utilities, which read the database map for many of the required parameters.

Example:

1. The application is `ciap` and the file in question is `voucher`. The database map is `ctc1prod` in the library `csdbmap` on the data volume. The `csgetaloc` statement is:

```
csgetaloc ciap voucher ctc1prod csdbmap data
```

Assuming the following database map entries:

Logical Library	Actual Library	Actual Volume	Logical File	Actual File
CIAP	CTRONSYS	&&CIAP		
CIAP	CTRON2	&&CIAP	VOUCHER	

the AP (application prefix `ciap`) files are owned by the database user `CTRONSYS`, with the exception of the logical file, `VOUCHER`, which is owned by the database user `CTRON2`. The `csgetaloc` query displays the following line:

```
VOUCHER CTRON2 &&CIAP
```

wname

The `wname` program is a method of reading the Computron LGMAP from the shell command line and extracting specific information, which must be specified in the command. It is used in some of the main Computron RDBMS utilities, most importantly the `csload_data` and `csunload_data` scripts. The program is located in the `$CTRON_EXEC/_ctron_` directory. The syntax is:

```
wname {LGMAP-name} [library] [file]
```

☞ *Note:* If the library and/or the file name is left blank, a blank must be provided encased in quotes in the following format: ‘ ‘

This program returns the full file system path associated with the LGMAP name provided. It can be used to read the LGMAP without running the application and is useful for troubleshooting many of the other utilities such as the `create_db` and `csload_data` utilities, which read the LGMAP for many of the required parameters.

☞ *Note:* If an LGMAP entry is invalid, this program returns the error: “BAD-VOL.!” However, this function does not validate the existence of the file system, directory or file, it only interprets the location based on the LGMAP entry.

Example:

1. The LGMAP entry is RDBMAP. The library and file name are not required The wname statement is:

```
wname rdbmap \ \ \ \
```

Assuming an LGMAP entry of “RDBMAP /ctron/exec/rdbmap/release,” the wname function shown above returns:

```
/ctron/exec/rdbmap/release
```

2. The LGMAP entry is DATA2. The library is csuvarm and the file is not required. The wname statement is:

```
wname data2 csuvarm \ \
```

Assuming an LGMAP entry of “DATA2 /ctron/data2,” the wname function shown above returns:

```
/ctron/data2/csuvarm
```

3. The LGMAP entry is EXEC2. The library and file name are not required The wname statement follows:

```
wname exec2 \ \ \ \
```

Assuming that the LGMAP does NOT contain the value EXEC2, the wname function shown above returns:

```
BAD-VOL!
```

changerec

The `changerec` program is a platform specific executable that can be used to manipulate sequential or text files. It can add carriage return characters, newline return characters or pad records with null. The parameters available for this program are:

-r

This parameter adds carriage return characters to records in a file.

-n

This parameter adds newline return characters to records in a file.

-z

This parameter pads records with null if the new record size is greater than the old record size.

The syntax for running the `changerec` program is:

```
changerec -{option} OLD-RECORD-SIZE NEW-RECORD-SIZE <old-file>new-file
```


6.4 Debugging Options

Introduction to Debugging Options

There are various logging techniques developed for troubleshooting the majority of issues that may be encountered in Computron RDBMS implementations. This section describes and provides instructions for enabling this logging.

Generic Steps for Debugging

Debugging produces very useful information. However, the following steps are recommended when attempting to generate a debug log with a logging variable that writes to the user's log file.

1. Export the debugging option(s) desired, either manually with the export command or using the dbdebug option.
2. Truncate the user log file for the user who generating the log. This can be done with the command: "`> {user-log-name}`". This causes the log file to be cleared, without removing the file.
3. If this logging is to be generated in a VB or WEBdesk presentation, follow the additional steps outlined below:
 - a) become root in order to start a csserve session
 - b) verify that the DEBUG variable is exported in the environment by typing in the following command from the shell command line:


```
"echo ${DEBUG-OPTION-NAME}"
```

Examples:

```
echo $CTRON_DEBUG_DB
```

```
echo $CTRON_DEBUG_IOCOUNTER
```

- c) start a new csserve session and have the user use this port number for the logging test only. This port performs with the logging enabled until it is killed. Any user(s) running from this port is writing this additional information to their log file and with certain types of logging, this can cause space issues on the server.
4. Once the log file has been created, the user should exit the Computron application completely, the log should be copied to another filename and the user's log file cleared.

 *Warning: If a VB or WEBdesk port was started for the logging, the users should only use this for testing because the log files could become very large.*

dbdebug

The dbdebug script is located in the file system \$CTRON_EXEC/db_run. It is run from the shell command line to export specific debugging variables to the environment. When the

debugging variables are set, they cause the runtime to output different types of processing information to the user's log file (refer to Section 6.7 of the *System Administrators Guide for Unix* for more information on the user log file).

Typing the command, `dbdebug`, provides a formatted list of the debugging options for the environment, along with a description of each.

```
Debugging environment variables for Oracle
Num  Setting Name          Description
 1   off  CTRON_DEBUG_DE        Turns on a lot of debugging info
 2   off  CTRON_DEBUG_IOCOUNT   Shows IO statistics
 3   off  CTRON_DEBUG_SQLTRACE  Create a trace file for db calls
 4   off  CTRON_DEBUG_DB_CONNECT Debug connecting to server
 5   off  CTRON_DEBUG_EVALUATERULES Show how rules were chosen
 6   off  CTRON_DEBUG_CSRANGE   Debug cstrange.c
 7   off  CTRON_DEBUG_SHOWSQL   Show sql statements
 8   off  CTRON_CSLOGHEX_BYTES  Number of CSLOGHEX bytes to print
 9   off  CTRON_DEBUG_TRANSACTION Generate transaction log
10   off  CTRON_DEBUG_XML       XML debugging
11   off  CTRON_DEBUG_USER_EXIT Runtime User Exit debugging
12   off  RUNCBL                Full path of runtime to use
Your selection:
Type number to toggle value, q quit, d scroll down, u scroll up, r refresh
```

Dbdebug options

As illustrated above, there are 8 commonly used debugging options. Some of these logging options overlap. Some of the options include the information provided in another type of logging. Refer to a more detailed description of the debug options later in this section.

In order to actually turn on debugging, the word `dbdebug` must be preceded by a period “.” followed by a space.

The syntax for `dbdebug` is:

```
. dbdebug
```

Once the list of the debugging options is displayed, enter the number corresponding to the desired option. The option then reflects the value `on`. Whenever the number of a debug option is entered, the value currently set is changed. For example, if the value is currently `on` and you enter the number again, the option changes to `off`. To exit the `dbdebug` window, enter “q”.

Enabling Database Debugging from Within Computron Applications

Debugging can be enabled from the Set Defaults for User Window (DS1) in character cell (PF-25 function key from a menu).

```

--- Set Defaults for User ---                               DS1
Enter Indicated Data and Press (RETURN)
--- or Press (PF-16) to Exit

Data Base Map:      EBPX10C                               in CSDBMAP                               on DATA

Default Company          Company scnio:          Batch Stream:
I/O Diagnostics      S (Off/Summary/Full)

Work/temporary files:   WORKDIR  WK006167   WORKVOL  WRKVOL
User programs:         RUNDIR                               RUNVOL  EXEC

--- For Submitted Jobs ---

Run or Hold  R (R/H)          Priority:  0
Queue Name:

RUNTIME=/enterprise/10qa/exec/_ctron_/wrunora,  REV=10.0.7.0
WISPGID=6167,  PID=6332,  NODENAME=supsun2,  OS=SunOS(5.10)
    
```

Set Defaults for User

I/O Diagnostics

This field turns on the level of debugging for this user. It is still advisable to truncate the user log file prior to generating the debug log. Valid entries are:

N – Off. Turn debugging off.

S – I/O statistic logging. This is equivalent to the shell command, `CTRON_DEBUG_IOCOUNTER`. When generating a log with this option, the log file is not updated until the program is exited and the user returns to the menu. Often administrators make the mistake of copying or trying to view the log when the user is still in the application and the information is not there because it hasn't been written yet. Since the program time is part of the I/O statistics, the longer the user remains idle within the application before returning to the menu, the greater the program time is.

F – Full database debug logging. This corresponds to the shell command, `CTRON_DEBUG_DB`

The table below provides a more detailed description of these options and information regarding when they might be used.

DEBUG OPTION	DESCRIPTION	COMMON USES
CTRON_DEBUG_D B	<p>This is the most comprehensive of the logging options in terms of the detail written to the user's log file. Among the information it prints are:</p> <ul style="list-style-type: none"> ◆ the application program being called, ◆ the logical files being accessed, ◆ the keys passed from the application and used by the runtime to access the data, ◆ the tables being opened, ◆ the opening mode and the method of access, ◆ the SQL statement generated and passed to the database, ◆ the data fetched from the database, ◆ the configuration settings of the Computron environment and certain configurations of the RDBMS system, ◆ other additional details. 	<p>Used in cases of analyzing a single process that is readily reproducible.</p> <p>Not recommended for a lengthy or data intensive process such as running reports or posting functions. This is more suited for inquiries, maintenance, or entry functions.</p> <p>Helpful in providing the smallest and most readable log,</p> <p>Computron recommends you truncate the log file, enable the debugging option, run the process and completely log out of the application before accessing the log file.</p>

DEBUG OPTION	DESCRIPTION	COMMON USES
CTRON_DEBUG _IOCOUNT	<p>This option logs statistics regarding data access. It includes:</p> <ul style="list-style-type: none"> ◆ the programs being run and the logical files being accessed. <p>For each logical file, it includes:</p> <ul style="list-style-type: none"> ◆ the reads, writes, re-writes, deletes and updates made in the database related to that file. ◆ It provides the number of rows fetched per second, ◆ information about the hits to the runtime cache. ◆ A summary of the total I/O time is provided for each logical file, as well as total program time. However, the program time is only valid if the program is exited immediately after processing is complete. 	<p>Used when performance is an issue.</p> <p>Helpful in determining where (in what process or table) the majority of the processing time is being spent, how much data is being read, etc.</p> <p>☞ <i>Note:</i> When generating a log for I/O statistics, the log file is not updated until the program is exited and the user returns to the menu. Also, since the program time is part of the I/O statistics, the longer the user remains idle within the application before returning to the menu the greater the program time statistic is.</p>
CTRON_DEBUG _SQLTRACE	<p>This option does not output information to the Computron user log. It enables a logging function within the RDBMS system which in turn generates a database specific file.</p> <p>☞ <i>Note:</i> In certain RDBMS systems, such as Oracle, the file output from the trace must then be input to another process to generate useable information.</p>	<p>Used in cases where the CTRON_DEBUG_DB log does not provide any useful information and processing within the RDBMS layer needs to be analyzed.</p> <p>It is usually used when working with Technical or Development personnel.</p>
CTRON_DEBUG _DB_CONNECT	<p>This option provides information about the RDBMS connection such as the database name, the server name, the database ID and the database password to the user log file.</p>	<p>Used in cases where users cannot connect to the database. Certain variables, such as the RDBMS system not being available, should be ruled out first.</p>

DEBUG OPTION	DESCRIPTION	COMMON USES
CTRON_DEBUG_EVALUATERULES	This option provides information regarding how specific tables were chosen based on the information passed in the key from the application and information in the RDBMap. The evaluate_rules logic generally applies only to multi-tables RDBMaps, i.e., cases where a single logical file in the product translates to more than one database table.	Useful in cases where it is suspected that the wrong table is being accessed, such as a table containing header records instead of the table with the trailer records. It is also useful in situations where custom RDBMaps are being used but the process is not returning the expected results. It is usually used when working with Technical or Development personnel.
CTRON_DEBUG_CSRANGE	This option provides information about the CSRANGE query sent by the program and the way it was interpreted by the runtime.	Used in cases where performance is an issue or unexpected or incorrect results are being returned to the application. It is usually used when working with Technical or Development personnel.
CTRON_DEBUG_SHOWSQL	This option provides a list of all SQL statements used in the process, without the additional information generated when using CTRON_DEBUG_DB.	It is used when you are only interested in the SQL queries being sent to the RDBMS system and information such as access mode, key of access, config values, etc are not required.
CTRON_DEBUG_CACHE_STATS	This option provides information about caching that was performed by the runtime, without the additional I/O statistics. Cache statistics are also provided in CTRON_DEBUG_IOCOUNTER	It is used when you are only interested in the results of caching, not in the I/O statistics. It is useful when configuring an application to take advantage of database caching when system memory may be a consideration.

The “set -x” variable

“set -x” is a command that can be added to a shell script that outputs to either the screen or a redirected file a line-by-line report of what is taking place in the script. It shows the variables read in the script, the values used and decisions made in evaluation statements and the general path the script takes. It is useful when trying to follow the logic of a script especially when the results of processing the script are not what were expected. The “set -x” command can be enabled in some of the scripts used in the Computron environment with the use of environment variables. This provides an alternative to manually editing the script to add the “set -x” command and also provides for the output being directed to the user’s log file.

☞ *Note:* Editing any Computron script is not supported.

CSBATCH_DEBUG – Background Processing Debugging

The variable `CSBATCH_DEBUG` can be set to cause the activity from background processing to be logged to the user's log file. This logging is often useful in cases where background processing is not functioning as expected. Refer to Section 6.7 of the *System Administrators Guide for Unix* for more information on the user log file.

The syntax for enabling the logging of the background processing activity is:

```
CSBATCH_DEBUG="set -x"  
export CSBATCH_DEBUG
```

One way to ensure that this is set correctly is to type:

```
echo $CSBATCH_DEBUG
```

If the variable is set, the shell should return the value:

```
set -x
```

Refer to the heading at the beginning of this section labeled “Generic Steps for Debugging” for the next steps in using this option.

CSPRINT_DEBUG – Print Processing Debugging

The variable `CSPRINT_DEBUG` can be set to cause the activity from print processing to be logged to the user's log file. This logging is often useful in cases where the printing process is not functioning as expected. Refer to Section 5.1, “Computron Server Configuration Management – ADMINTOOLS,” of the *System Administrators Guide for Unix* for more information on the user log file.

The syntax for enabling the logging of print processing is:

```
CSPRINT_DEBUG="set -x"  
export CSPRINT_DEBUG
```

A method to ensure that this is set correctly is to type:

```
echo $CSPRINT_DEBUG
```

If the variable is set, the shell should return the value:

```
set -x
```

Refer to heading the beginning of this section labeled “Generic Steps for Debugging” for the next steps in using this option.

SCRIPT_PATH – various scripts

The variable `SCRIPT_PATH` can be set to cause the activity from a number of utilities to be logged to the user's log file. Refer to Section 6.7 of the *System Administrators Guide for Unix* for more information on the user log file.

Some of the utilities that evaluate the SCRIPT_PATH variable are:

- ◆ cscron
- ◆ csdb_install
- ◆ update_dbfiles
- ◆ csload_data
- ◆ csunload_data
- ◆ rdbmap_diff

The steps for processing with the SCRIPT_PATH method of logging are slightly different than with the script logging, since the scripts listed above are not run from within the Computron application and these processes cannot be run in an AUI session (VB or WEBdesk).

In this case, the following process should be used:

1. Export the debugging option(s) desired.
2. Truncate the user log file for the user who will be generating the log. This can be done with the following command: > {user-log-name}. This causes the log file to be cleared, without removing the file.
3. Run the script or process that calls the script. Remember to exit out of the application completely.
4. Review the log file.

The syntax for enabling the SCRIPT_PATH method of logging is:

```
SCRIPT_PATH="set -x"  
export SCRIPT_PATH
```

A method to ensure that this is set correctly is to type:

```
echo $ SCRIPT_PATH
```

If the variable is set, the shell should return the value:

```
set -x
```


6.5 RDBMap Format Maintenance


Overview of RDBMAP Format Maintenance

The Computron RDBMaps serve as an external schema that defines the tables in which Computron data is stored. The RDBMaps are used constantly by the system to determine the location of the required data.


Computron software identifies required data by a logical name. The database map directs the system to the physical location of the corresponding data. Relational database implementations require an additional step in the data access process. The database map contains the name of the relational database, the relational database management system (RDBMS) server and the location of the RDBMap file that corresponds to the logical file name that is requested by the application. The RDBMap, in conjunction with the RDBMS, provides the application with the physical location of the required data.

Introduction of RDBMAP Format Maintenance

Before maintaining an RDBMap, you must know the logical file names used by Computron applications. Computron applications use Data Description Lists (DDLs) to identify data. DDLs are used as the name of data fields on screens, windows and reports, as well as the location for the data in a relational database. Information such as logical name, length and format is contained in the DDL. The RDBMap format is a liaison between the Computron applications (thus, the DDLs) and the relational database. The RDBMap files describe the data stored in Computron tables to the applications and the data passed from the application to the RDBMS.

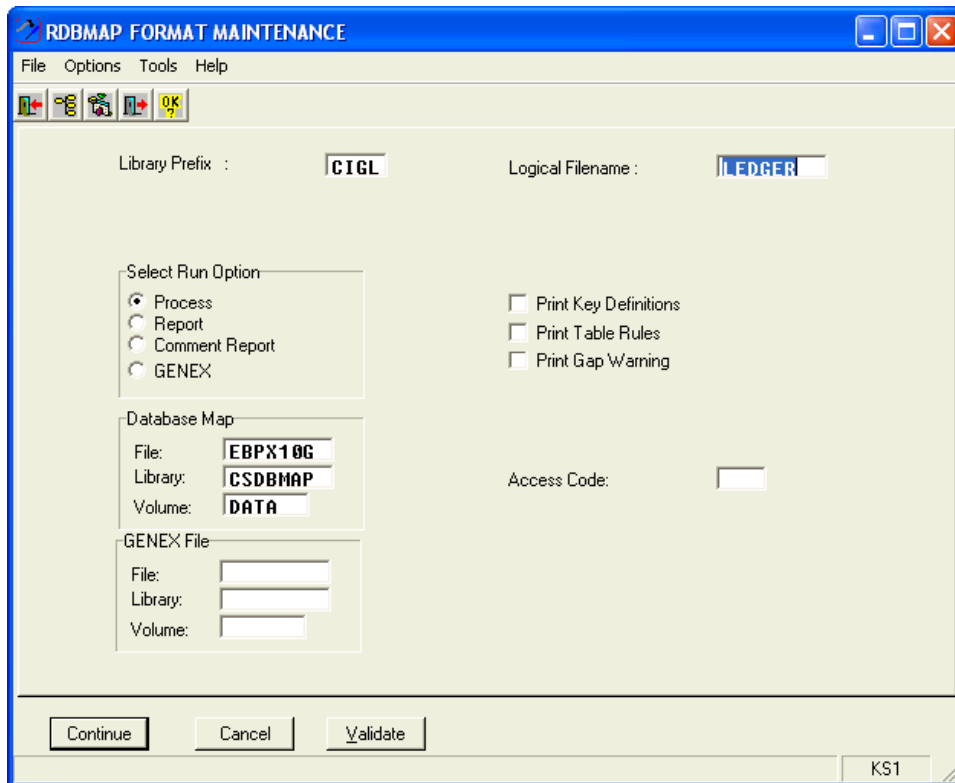
 *Note:* The DDLs used by each Computron application are described in detail in the corresponding application's Technical Guide.

Computron creates RDBMaps for the files in each application using the Create from copw or Modify from copw buttons. Creating RDBMaps should not be done by clients.

 *Warning:* Clients should only modify the User Interface Area of RDBMaps. The other columns are the responsibility of Computron.

RDBMAP Format Maintenance Window (KS1)

The RDBMAP Format Maintenance Window (KS1) identifies the RDBMap that you want to maintain.



RDBMAP Format Maintenance Window (KS1)

Library Prefix

Enter the Computron system prefix for the product associated with the RDBMap to maintain. For example, ciap, cigl, csuv, etc.

Logical Filename

Enter the name of the logical file associated with the RDBMap to maintain.

Select Run Option

Enter the operation to perform. Valid options are:

Process – This option enables you to view the requested RDBMap online and modify the different records connected to this file.

Report – This option produces a printed report that contains information about the RDBMap. Information such as field definitions, key definitions, table definitions and gap warnings are included on the report. The following three fields can be selected to add additional information to the report. A single report in the user's print queue is made up of several RDBMap files when more than one RDBMap Report request is made before exiting the RDBMAP Format Maintenance function.

Comment Report – This option is currently not available.

GENEX – This option allows you to GENEX (Generalized External Interface) or import an RDBmap into the Computron system. Enter the name of the new XMLMap in the XML Filename field. This option is used with the GENEX File: File, Library and Volume fields, defined later.

⚠ *Warning: This option should only be used with Computron supervision.*

Default: Process

The following three fields apply only when the Select Run Option field is set to Report.

Print Key Definitions

Include in the report the Key record information. Key records describe the table indexes. Valid options are:

- Print the Key record type details on the report.
- Do not include Key record information on the report.

Default:

Print Table Rules

Include in the report the Table record information. Valid options are:

- Print the Table record type information on the report.
- Do not print the Table record information on the report.

Default:

Print Gap Warning

Include on the report messages regarding discontinuity of defined fields, i.e., gap warning. Valid options are:

- Print gap warning messages on the report.
- Do not print gap warning messages on the report.

Default:

Database Map File: / Library: / Volume:

Enter the name and location of the database map that identifies the location of the RDBMaps.


Access Code

This field is currently not implemented.

GENEX File File: / Library: / Volume:

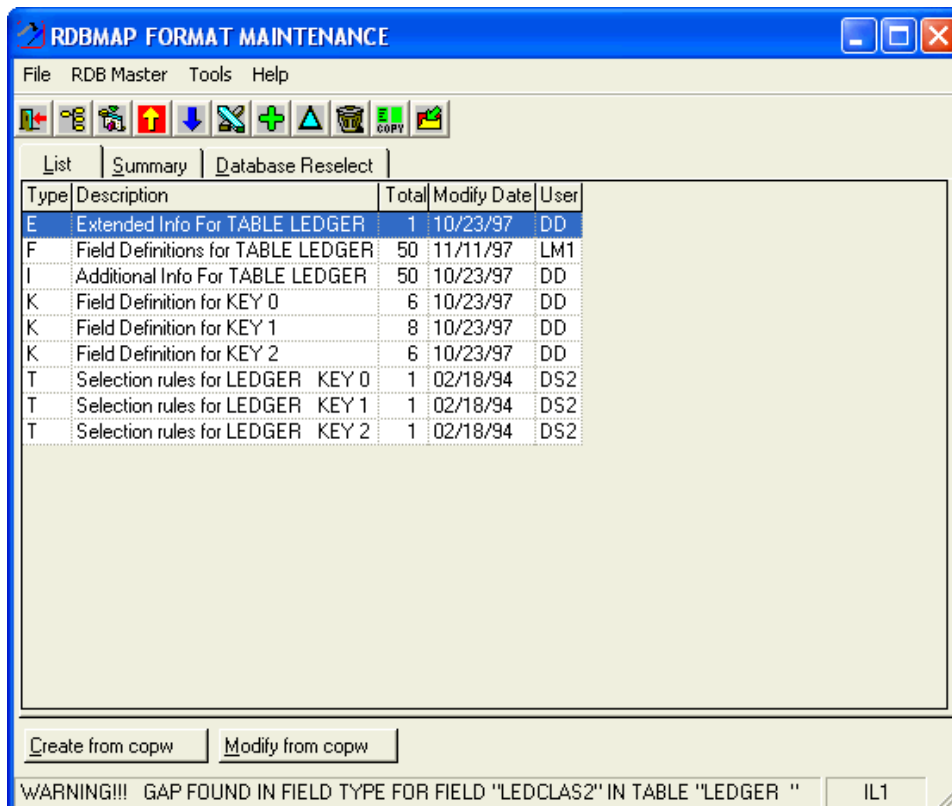
Enter the name and location of the input file to create an RDBMap. This is used when you select the GENEX option of the Select Run Option field. The file can be in either XML or flat file format. The format of the file is defined by the RDBGEX COPW and the rdbgenex DDL or the rdbgenex XML map. The file layout provides for the following record types:

Record Type	Description
P	Parameter record
T	Table record
K	Key information record
F	Field information record
E	Extended record

 *Warning: These fields should only be used with Computron supervision.*

RDBMAP Format Maintenance Window (IL1)

To open the IL1 window, select the Process option for the Select Run Option field and then click Continue.



RDBMAP Format Maintenance Inquiry/List Window (IL1)

On the Summary tab, the system displays the filename, library and volume based on the information entered into the Library Prefix, Logical Filename and Database Map fields on the KS1 window. These fields are non-modifiable; however, it is important to verify that the RDBMap(s) listed is the one that you intend to maintain or print.

To display a record, double click on it or highlight it and select one of the icons described in the table below.

Type

There are five types of records associated with each RDBMap.

E – These records link logical file names to table names used by the relational database.

F – These records contain information connecting DDL field names and the columns in the table.








I – These records contain information connecting internal development information (COBOL names) and Column names.





K – These records contain index definitions.

T – These records contain rules that the application applies to locate the table where the data is for the indexes.

To process your entry, use one of the following options.

Note: Computron applications' RDBMaps should only be modified under Computron Client Support supervision.

Computron Options — RDBMAP Format Maintenance Inquiry/List Window	
Button	Description
	Exit – This option exits the Processing function of RDBMap Format Maintenance.
	Previous – Returns to the previous window without exiting the RDBMap function.
	More Data – Loads additional records in the UL1 window.
	Create new records for the current RDBMAP file.
	Modify existing records.
	Removes existing records.
	Creates a new record by duplicating an existing record. You then modify it to form a unique record.

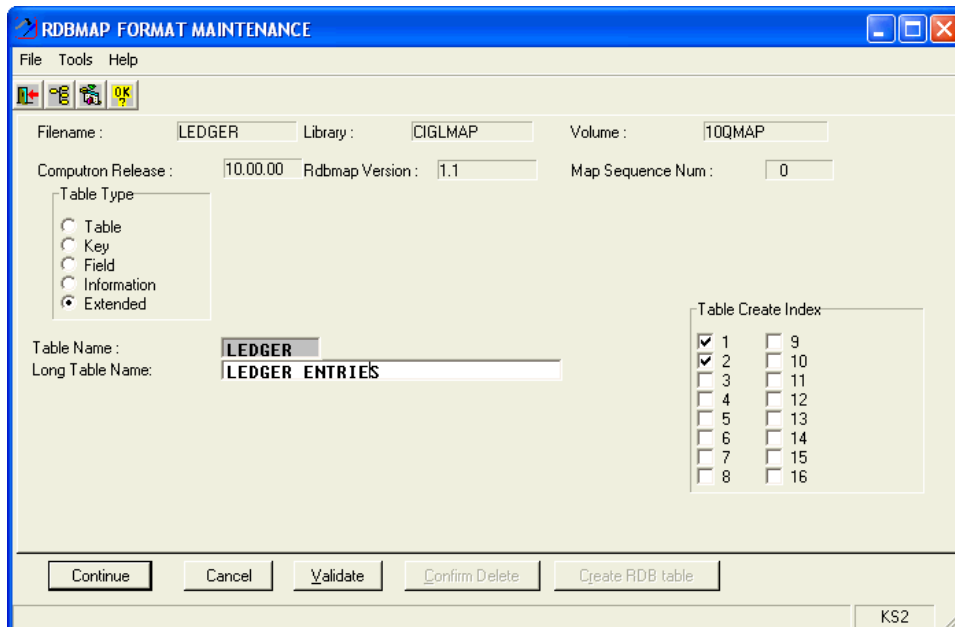
Computron Options — RDBMAP Format Maintenance Inquiry/List Window	
Button	Description
	<p>Save – Fully implements the modifications to the RDBMap. Specifically, this option verifies that table records exist for all keys identified via the Table Create Index check boxes on the KS2 window for Field records; if corresponding table rules do not exist, the appropriate checkbox is cleared (<input type="checkbox"/>) for that key. Also, where the Known As Name field is blank (Field records and Key records), this option populates the associated DDL name into the field. Click this button after performing any of the other functions to save changes to the RDBMap file.</p>
	<p>Creates new Key, Field, Info and Extend records for the RDBMap based on the file layout directories contained in the current Computron software environment (CIXXCOPW and CIXXCOPY [where CIXX represents the system prefix entered on the RDBMAP Format Maintenance Window (KS1)]).</p> <p> <i>Caution! This option replaces any existing records. Make a backup of the existing RDBMap file prior to completing this function.</i></p> <p><i>Validations:</i> The following validations are performed upon selection of this function:</p> <p>This function verifies that the field type records account for all bytes in the table row and no gaps exist between the end of a defined field and the beginning of the next field. Any discontinuity results in a Gap Warning message on the window.</p> <p>Defined fields are ordered by starting position in the row. If fields are sequenced out of order, an error message is displayed and the user must use the Modify from copw option to resort the fields.</p>
	<p>Modifies existing Key, Field, Info and Extend records or creates new records where appropriate, for the RDBMap based on the file layout directories contained in the current Computron software environment.</p> <p><i>Validations:</i> This function verifies that the field type records account for all bytes in the table row and no gaps exist between the end of a defined field and the beginning of the next field. Any discontinuity results in a Gap Warning message on the window.</p>

The Headings that follow describe each record type in detail.

Note: Clicking the Add button prompts you for the data required to create all types of records simultaneously. Since the discussions that follow are specific to each record type, the Change button was selected to capture window illustrations because only the applicable fields are displayed on each KS2 window in this mode.

Extended (E) Records

The Extended (E) record associates the logical file name with the Table Name (up to eight characters) and the Long Table Name where the application's data is stored. This information is created by Computron.



RDBMAP Format Maintenance Window (KS2) – Extend Records

Table Type

For an extended record, this is Extended (E).

Table Name

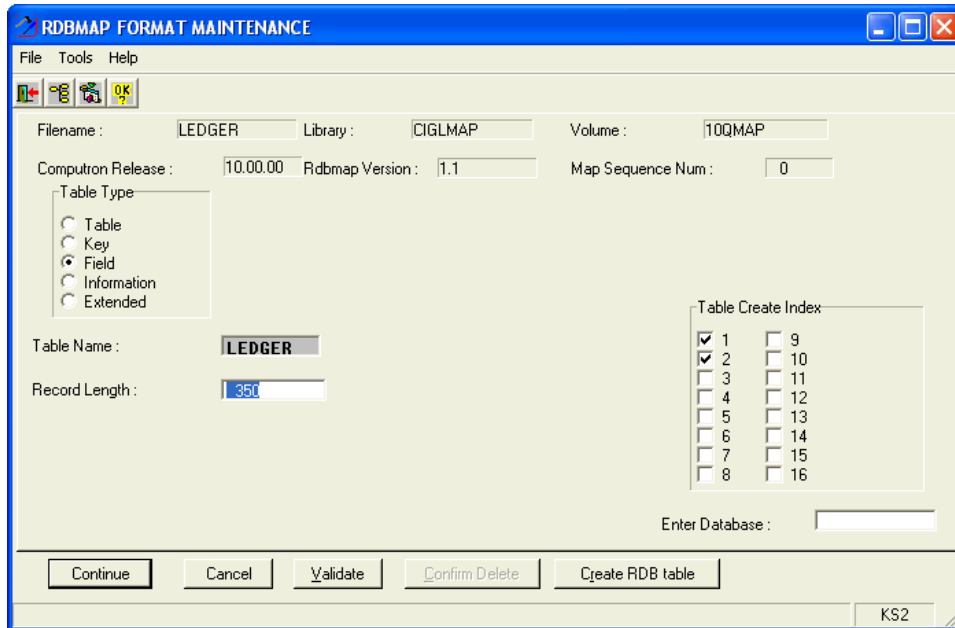
This is the eight character short table name used internally by the application and used in all other records of the RDBMap files. It provides a short-hand alias for the actual table name used in the RDBMS.

Long Table Name

This is the table name that the RDBMS uses to identify the table (maximum 18 characters). If this field is left blank, the short table name (eight characters) is used.

Field Records

Field records establish the columns that compose the table. A single table can contain up to 250 columns. Additionally, the key records that are to be used when creating indexes for this table are identified via field records.



RDBMap Format Maintenance Window (KS2) – Field Records

Table Type

For a field record, this is Field (F).

Table Name

This is the eight character (short) name that refers to this table.

Record Length

This is the total number of bytes in each row of this table.

Table Create Index

This identifies whether the index that corresponds to each possible alternate key should be created for this table. Valid entries are:

- This key is in use.
- This key is not in use.

Default:

Note: If a Table record (T) doesn't exist for a corresponding selected key, the box is then cleared () upon saving the RDBMap changes from the RDBMap Format Maintenance Inquiry/List Window (IL1).

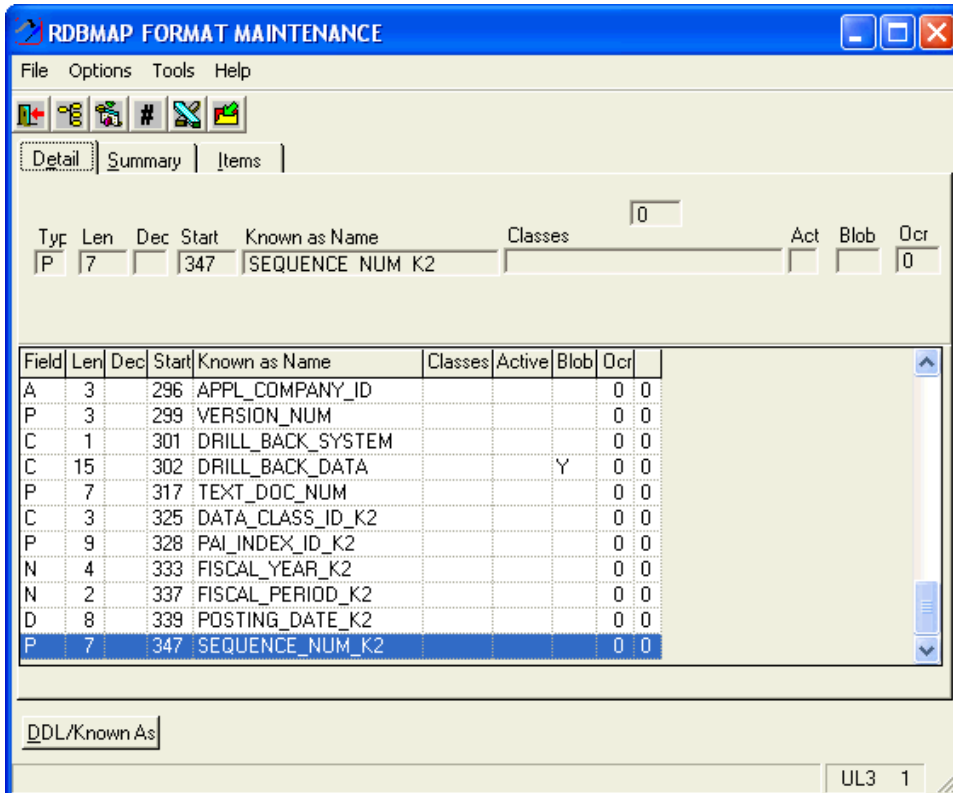
Enter Database:

This field is currently not implemented.

Create RDB table

This button is currently not implemented.

The format details of each column are displayed on the UL3 window.



The screenshot shows the 'RDBMAP FORMAT MAINTENANCE' window with the 'Detail' tab selected. A table displays field records with columns: Field, Len, Dec, Start, Known as Name, Classes, Active, Blob, and Ocr. The record for 'SEQUENCE_NUM_K2' is highlighted in blue.

Field	Len	Dec	Start	Known as Name	Classes	Active	Blob	Ocr
A	3		296	APPL_COMPANY_ID				0 0
P	3		299	VERSION_NUM				0 0
C	1		301	DRILL_BACK_SYSTEM				0 0
C	15		302	DRILL_BACK_DATA		Y		0 0
P	7		317	TEXT_DOC_NUM				0 0
C	3		325	DATA_CLASS_ID_K2				0 0
P	9		328	PAI_INDEX_ID_K2				0 0
N	4		333	FISCAL_YEAR_K2				0 0
N	2		337	FISCAL_PERIOD_K2				0 0
D	8		339	POSTING_DATE_K2				0 0
P	7		347	SEQUENCE_NUM_K2				0 0

RDBMAP Format Maintenance Window (UL3) – Field Records

Field

This is the type of data in this column. This data type corresponds to the format defined in the DDL of the associated data.

Note: The field types correspond to relational data types differently for each RDBMS with which Computron's software is interfaced. Refer to the appropriate appendix at the end of this guide for a complete listing of the field type values and the corresponding data types.

Len

This is the number of bytes this column occupies.

Dec

For numeric and decimal database data types, this is the number of digits accepted to the right of the decimal point.

Start

This is the position in the application buffer where the data for this column begins.

Known as Name

This is a descriptive name of the column name of this table (18 characters maximum). It corresponds to the DDL name. The Known As Name must coincide with the Known As Name entered in the key definition record. If left blank, the system applies the DDL name to this field value upon saving changes from the RDBMap Format Maintenance Inquiry/List (IL1) window.

Classes

This field is currently not implemented.

Active

This indicates the status of this column. Valid entries are:

Y/Blank – Indicates the column is defined in the table.

N – Indicates the column is not defined in the table.

Blob Switch

Binary Large Object (BLOB) Indicator field. A BLOB is a database type that is most often used to represent indiscriminate data.

Occ

This column only applies in XMLMap Maintenance.

When you select the DDL/Known As button, the RDBMAP Field Format Maintenance Window (UL3) displays. On this window, more information about each column displays.

DDL Name

This is the name of the field from the DDL.

Ordinal

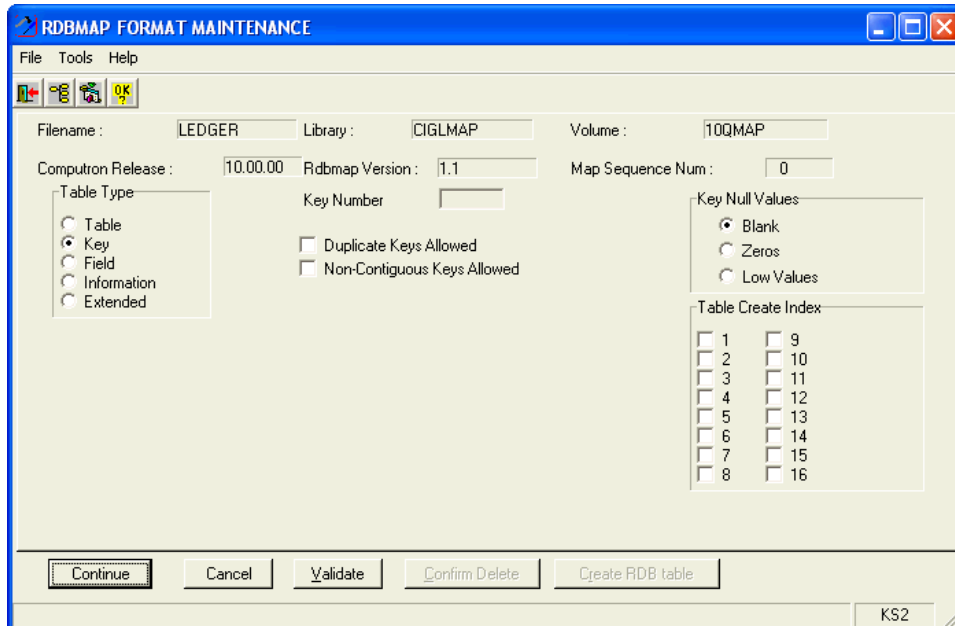
This column is not in use.

Virtual

This identifies a column as a virtual column.

Key (K) Records

Key records define the indexes for a table. The information is taken from the key definitions in the DDL.



RDBMAP Format Maintenance Window (KS2) – Key Records

Table Type

For a Key record, this field is Key (K).

Key Number

This is the table index being defined. Valid values are 0–16, where 0 (which does not display) designates the primary index and 1–16 represent secondary indexes. The primary index corresponds to the primary key in the DDL. The secondary indexes correspond to the alternate keys in the DDL.

Duplicate Keys Allowed

This field determines whether duplicate values are allowed for this index. Valid options are:

- Index is not unique. Duplicate values are allowed.
- Index is unique. Values in the index **must not** be duplicated.

Note: Duplicates are NOT allowed for the primary index of a table.

Default:

Non-Contiguous Keys Allowed

This indicates whether the data columns that make up this index are adjacent to each other. Valid options are:

- The data columns that make up the index are not contiguous.

- The data columns that make up the index are contiguous.

Default:

Key Null Values

This indicates what value is passed by the application to force the index columns to be stored as null. Valid options are:

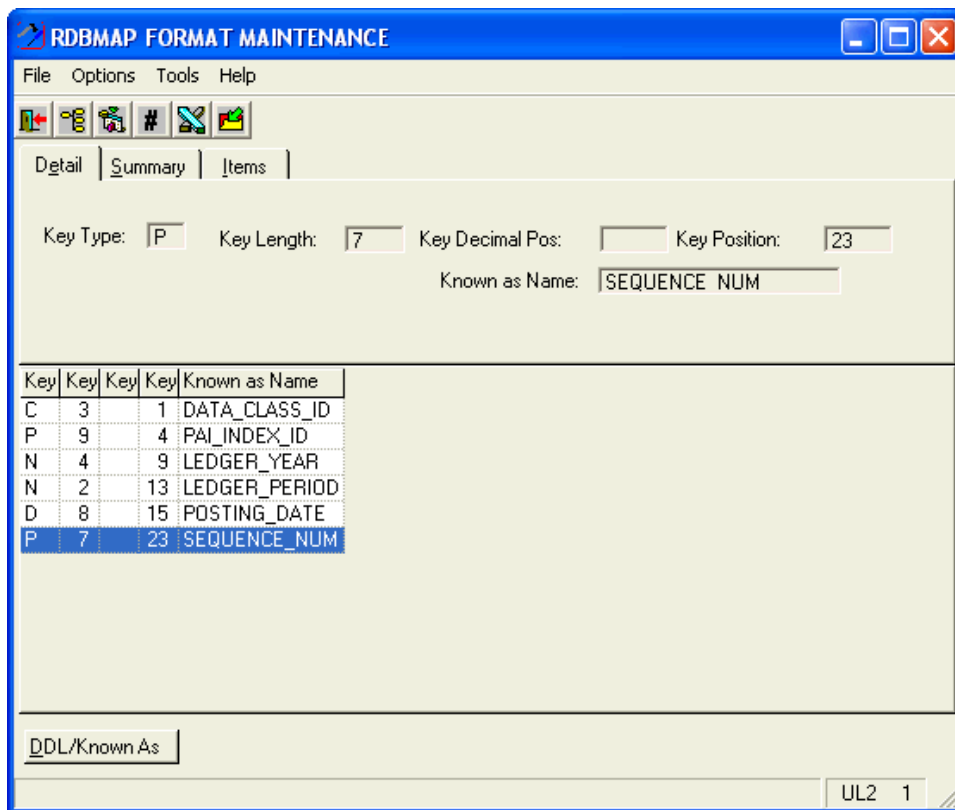
Blank – Indicates a value containing all spaces is considered null.

Zero – Indicates a value containing all zeros is considered null.

Low Values – Indicates a value containing low values is considered null.

Default: Blank.

The UL2 window identifies the columns that comprise the index to the table.



RDBMAP Format Maintenance Window (UL2) – Key Records

Key Type

This is the type of data within the column. Types include:

- ◆ C – Character
- ◆ A – Alphanumeric
- ◆ N – Numeric
- ◆ D – Date

◆ P – Packed

☞ *Note:* The key types correspond to relational data types differently for each RDBMS with which Computron's software interfaces. Refer to your database administrator for a complete listing of the key field type values and the corresponding data types.

Key Length

This is the number of bytes this column occupies.

Key Decimal Pos

For numeric and decimal database data types, this is the number of digits accepted to the right of the decimal point.

Key Position

This is the position in the application buffer where the data for this index begins.

Known as Name

This is a descriptive name for the DDL name that the RDBMS uses as the column name (18 characters maximum). The Known as Name entered when defining the index must coincide with the Known as Name entered in the Field record (F).

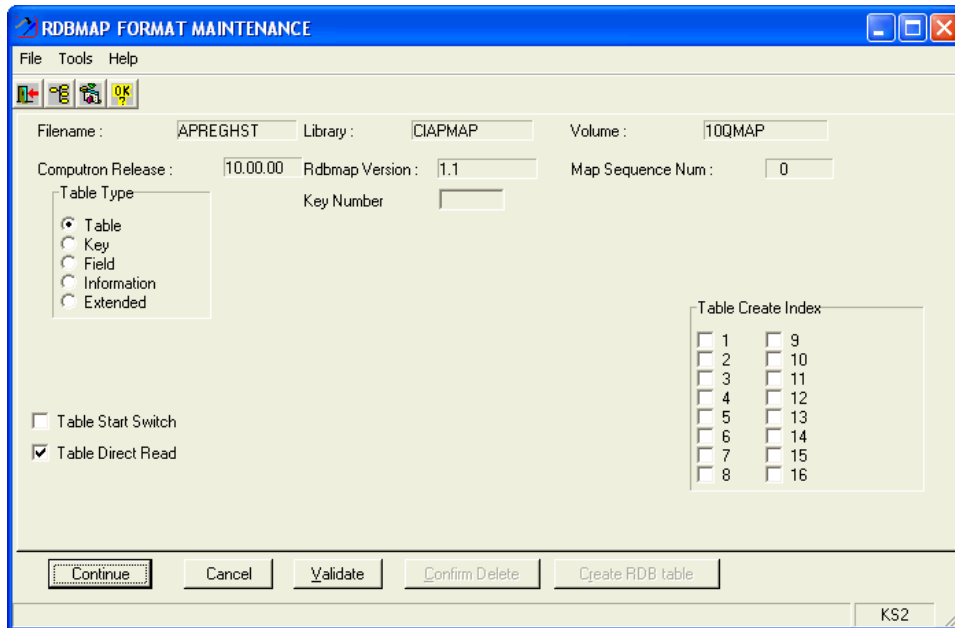
When you select the DDL/Known As button, the RDBMAP Format Maintenance Window (UL2) displays. On this window, more information about each column displays.

DDL Name

This is the DDL name for the column.

Table Rules

Table rules are used by the system to determine the name of the table where the data required by the application is stored. Some DDLs describe more than one record type. When this occurs, the RDBMap contains multiple Table records. When the application accesses the RDBMap via a key, these table rules identify which Table the application is addressing. For example, the AP file, APREGHST, contains a header (H) and a detail (L) record. In RDBMap Maintenance, the header record is identified as table, APRGHSTH, and the detail record is identified as table, APRGHSTL. These rules, tell the application which table to update or search.



RDBMAP Format Maintenance Window (KS2) – Table Records

Table Type

For a Table Rules record, this is Table (T).

Key Number

A number zero (0) through sixteen (16), identifies the table index. The primary key is key number 0 and does not display.

Table Start Switch

The Table Start Switch implies that the rules defined by this table record are to be evaluated. A specific table is used to retrieve or receive the data when the application requires a set of data. Valid options are:

- Evaluate the rules to determine a single table for the data.
- Extract data from all tables associated with the logical file name and perform a sort/merge to identify the order of rows.

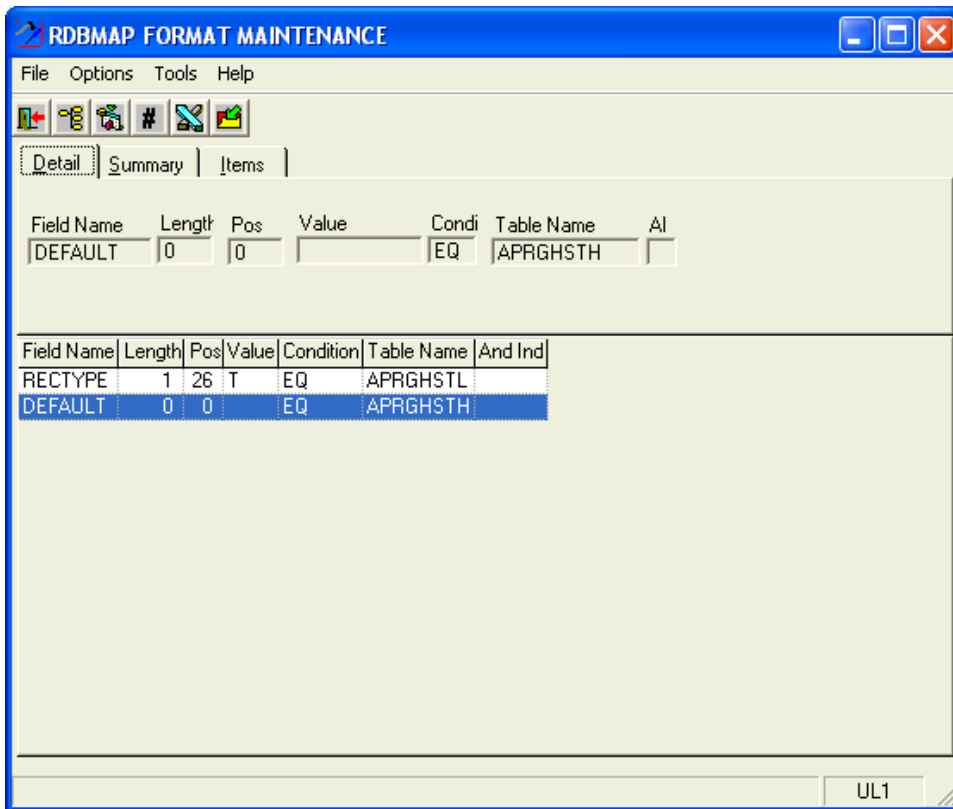
Table Direct Read

The Table Direct Read field indicates that the rules defined by this table record are to be evaluated. A specific table is used to retrieve or receive the data when the application requires a single data value. Valid options are:

- Evaluate the rules defined in this table record.
- Extract data from all tables associated with the logical file name.

Note: The Table Start Switch and Table Direct Read field values can be overridden by the applications.

The rules that the application follows in order to select the table containing the required data are defined on the UL1 window.



RDBMAP Format Maintenance Window (UL1) – Table Records

Field Name

This is the Data Description List (DDL) field name whose value is used to establish a table rule. When “Default” is the value in this field, an “else” condition is being established. For example, the APREGHST file table rule for the primary key is

```
RECTYPE 1 26 T EQ APRGHSTL
DEFAULT 0 0 EQ APRGHSTH
```

is interpreted as “If the field RECTYPE equals T use the APRGHSTL (detail) table else use the APRGHSTH (header) table.”

Length

Enter the number of bytes to compare when evaluating this rule.

Pos

Enter the position, within the application buffer, where the value of this DDL field is found. This position field **must** correspond to the Key Position indicated in the Key Record that defines the selected key.

Value

Enter the value to be used for comparison with the value of the DDL field passed by the application buffer.

Condition

Enter an operator used for comparing the value in the rule with the value passed by the application buffer. Valid entries are:

EQ – This operator represents the rule “Value must equal the value passed by the application buffer”.

NE – This operator represents the “Value indicated in the rule is not equal to the value passed by the application buffer”.

Table Name

Enter the eight character (short) name of the table.

And Ind

This field is currently not implemented.

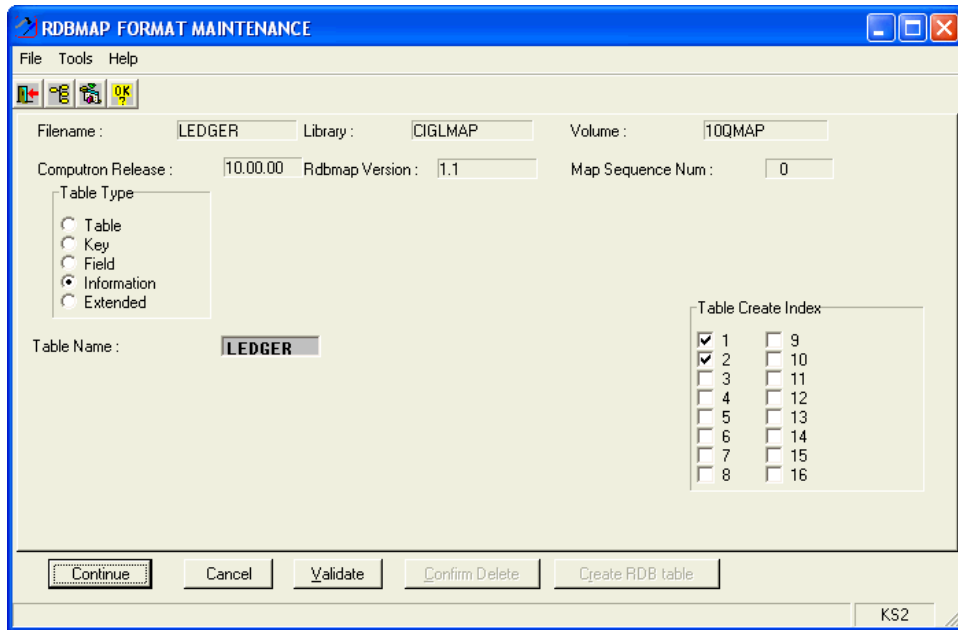
The Default Table Rule

All tables must be defined with a rule that is used when none of the rules satisfy the needs of the application. This rule is always defined as the last rule in the Table record and contains the following field values:

Field Name	Default
Length	0
Pos	0
Value	Blank
Condition	EQ
Table Name	Name of table to be used when no other rules is appropriate.

Info Records

Information records link the table name and its column names to the internal development names identified in Computron’s applications and COPW file layouts. This record is used primarily for associating the internal development identification and DDL information with the tables defined in the RDBMap file.



RDBMAP Format Maintenance Window (KS2) – Info Records

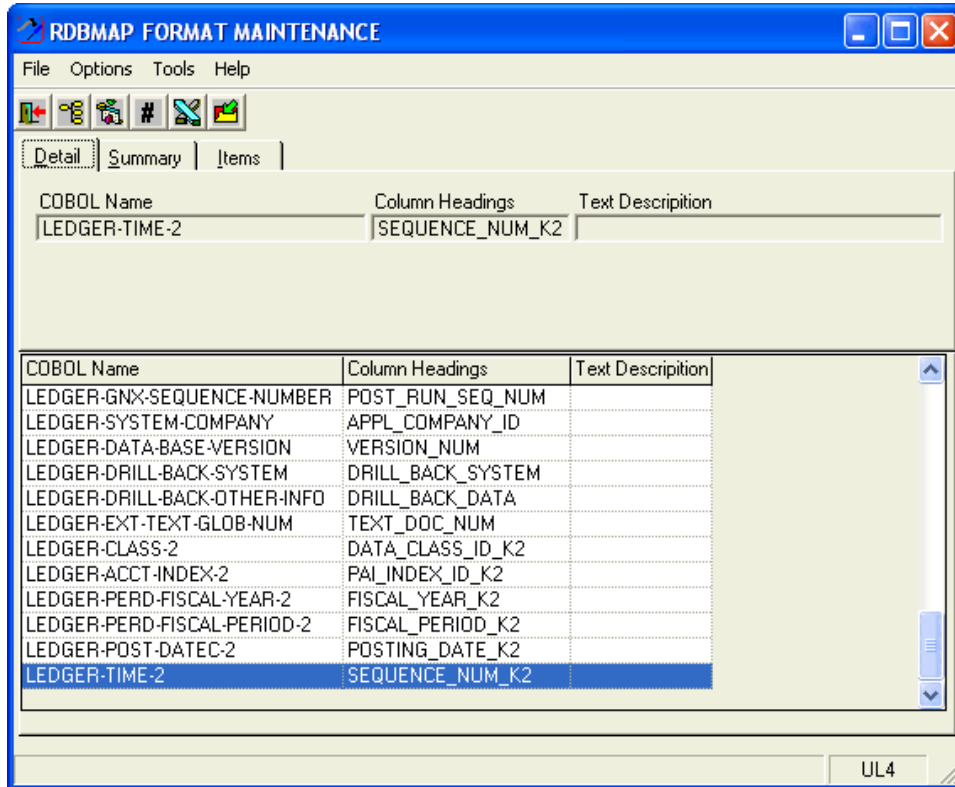
Table Type

For Information records, this is Information (I).

Table Name

The eight-character table name associated with the logical file name identified by the application.

The UL4 window connects Computron’s internal development name and the Known as Name of each column.



RDBMAP Format Maintenance Window (UL4) – Info Records

COBOL Name

Enter the actual, hard-coded reference to values in this column of the table. These values are identified in the COPW file layouts.

Column Headings

Enter the Known as Name that the RDBMS uses as the column heading of this table (18 characters maximum). The Column Heading entered must coincide with the Known as Name entered in the field and key definition records.

Text Description

Enter a description (up to 30 characters) that describes this column.

Report Option

A sample RDBMap Format report is shown here:

```

*   Computron Software, LLC   ***** Mon May 21, 2007 11:10 AM *
*
*                               RDBMAP Report for Schema LEDGER
*                               Column Definitions
*
*----- Page 1 *

```

Table Name : LEDGER LEDGER_ENTRIES

Record Length 350

DDL Name	DBMS Known As Name	DB Type	Length	Active Switch
LEDCLASS	DATA_CLASS_ID	CHAR	3	
INDEX0	PAI_INDEX_ID	NUMBER	9	
PERDCCYY	LEDGER_YEAR	NUMBER	4	
PERDPP	LEDGER_PERIOD	NUMBER	2	
POSTDTC	POSTING_DATE	DATE	8	
GLOBNUM	SEQUENCE_NUM	NUMBER	7	
COMPNUM	COMPANY_ID	CHAR	3	
CURRENCY	CURRENCY_ID	CHAR	4	
JOURNAL	JOURNAL_ID	CHAR	3	
PERDCYD	FISCAL_YEAR	NUMBER	4	
PERDPPD	FISCAL_PERIOD	NUMBER	2	
BATCHNUM	BATCH_NUMBER	CHAR	9	
TRANSNUM	TRANSACTION_NUMBER	CHAR	9	
DETLNUM	LINE_NUM	NUMBER	5	
TREF1	HDR_REFERENCE1	CHAR	30	
TREF2	HDR_REFERENCE2	CHAR	8	
TREF3	HDR_REFERENCE3	CHAR	12	
TREF4	HDR_REFERENCE4	CHAR	10	
TREF5	HDR_REFERENCE5	CHAR	15	
TRIFACE	USER_INTERFACE	CHAR	30	N
TRANDTCT	TRANSACTION_DATE	DATE	8	
SRCTRNUM	SRC_TRANS_NUMBER	CHAR	9	
SOURCECO	SOURCE_COMPANY_ID	CHAR	3	
LDGERAMT	LEDGER_AMT	DECIMAL	15, 2	
DETLREF	LINE_REFERENCE1	CHAR	30	
LDGPODRC	POSTING_CCY_ID	CHAR	4	
LDGDLCUR	DUAL_CCY_ID	CHAR	4	
LDGEROTH	OTHER_CCY_AMT	DECIMAL	15, 2	
LEDUNITS	LEDGER_UNITS	DECIMAL	15, 2	
CREDATE	CREATE_DATE	DATE	8	
CREATEID	CREATE_USER_ID	CHAR	3	
LDGERCRP	CORPORATE_AMT	DECIMAL	15, 2	
LDGDCTYP	POSTING_TYPE_IND	CHAR	1	
LDGTRDRC	TRANS_CCY_ID	CHAR	4	
LDGTRCUR	TRANS_PRIME_CCY_ID	CHAR	4	
LDGOTHAT	OTHER_AMT_IND	CHAR	1	
LDGDUAT	DUAL_AMT_IND	CHAR	1	
LDGRTDTC	CURRENCY_RATE_DATE	DATE	8	
LDGRSEQ	POST_RUN_SEQ_NUM	NUMBER	9	
LDGSYSO	APPL_COMPANY_ID	CHAR	3	
VERSION	VERSION_NUM	NUMBER	3	
DBACKSYS	DRILL_BACK_SYSTEM	CHAR	1	
DBACKKEY	DRILL_BACK_DATA	CHAR	15	

```

*   Computron Software, LLC   ***** Mon May 21, 2007 11:10 AM *
*
*                               RDBMAP Report for Schema LEDGER
*                               Column Definitions
*
*----- Page 2 *

```

DDL Name	DBMS Known As Name	DB Type	Length	Active Switch
TEXTGLOB	TEXT_DOC_NUM	NUMBER	7	
*** WARNING !!! GAP FOUND ***				
LEDCLAS2	DATA_CLASS_ID_K2	CHAR	3	
INDEX2	PAI_INDEX_ID_K2	NUMBER	9	
PERDCCY2	FISCAL_YEAR_K2	NUMBER	4	
PERDPP2	FISCAL_PERIOD_K2	NUMBER	2	
POSTDTC2	POSTING_DATE_K2	DATE	8	
LDGTIME2	SEQUENCE_NUM_K2	NUMBER	7	

```

*   Computron Software, LLC   ***** Mon May 21, 2007 11:10 AM *
*
*                               RDBMAP Report for Schema LEDGER
*                               Index Definitions
*
*----- Page 3 *

```

Index Number: 0 Index Null Value: Index Dups Allowed: N Non Contiguous Allowed:

DDL Name	DBMS Known As Name	DB Type	Length
LEDCLASS	DATA_CLASS_ID	CHAR	3
INDEX0	PAI_INDEX_ID	NUMBER	9
PERDCCYY	LEDGER_YEAR	NUMBER	4
PERDPP	LEDGER_PERIOD	NUMBER	2
POSTDTC	POSTING_DATE	DATE	8
GLOBNUM	SEQUENCE_NUM	NUMBER	7

Index Number: 1 Index Null Value: B Index Dups Allowed: Y Non Contiguous Allowed:

DDL Name	DBMS Known As Name	DB Type	Length
COMPNUM	COMPANY_ID	CHAR	3
CURRENCY	CURRENCY_ID	CHAR	4
JOURNAL	JOURNAL_ID	CHAR	3
PERDCYD	FISCAL_YEAR	NUMBER	4
PERDPPD	FISCAL_PERIOD	NUMBER	2
BATCHNUM	BATCH_NUMBER	CHAR	9

```

TRANSNUM  TRANSACTION_NUMBER  CHAR      9
DETLNUM   LINE_NUM             NUMBER    5
    
```

Index Number: 2 Index Null Value: B Index Dups Allowed: N Non Contiguous Allowed:

DDL Name	DBMS Known As Name	DB Type	Length
LEDCLAS2	DATA_CLASS_ID_K2	CHAR	3
INDEX2	PAT_INDEX_ID_K2	NUMBER	9
PERDCY2	FISCAL_YEAR_K2	NUMBER	4
PERDPP2	FISCAL_PERIOD_K2	NUMBER	2
PSTDTC2	POSTING_DATE_K2	DATE	8
LDGTIME2	SEQUENCE_NUM_K2	NUMBER	7

```

*      Computron Software, LLC      ***** Mon May 21, 2007 11:10 AM *
*
*                                RDBMAP Report for Schema LEDGER
*                                Table Definitions
*
***** Page      4 *
    
```

Index Number: 0 Table Start switch: Y Table Dread switch: Y

Field Name	Length	Pos	Value	Condition	Table Name	And Ind
		1		EQ	LEDGER	

Index Number: 1 Table Start switch: Y Table Dread switch: Y

Field Name	Length	Pos	Value	Condition	Table Name	And Ind
		1		EQ	LEDGER	

Index Number: 2 Table Start switch: Y Table Dread switch: Y

Field Name	Length	Pos	Value	Condition	Table Name	And Ind
		1		EQ	LEDGER	

6.6 XMLMAP Format Maintenance

Overview of XML and Computron

The Computron XMLMaps serve as external schemas that define the required structure of XML files. The XML files are used by Computron's Generalized External Interface (GENEX) for importing data into the Computron data repository or exporting data from the repository. Prior to the introduction of the XML technology, GENEX interfaces were available only for files in the flat file format. The XMLMap is used as a schema to allow files in .xml format to be input into the Computron data repository using the GENEX functions.

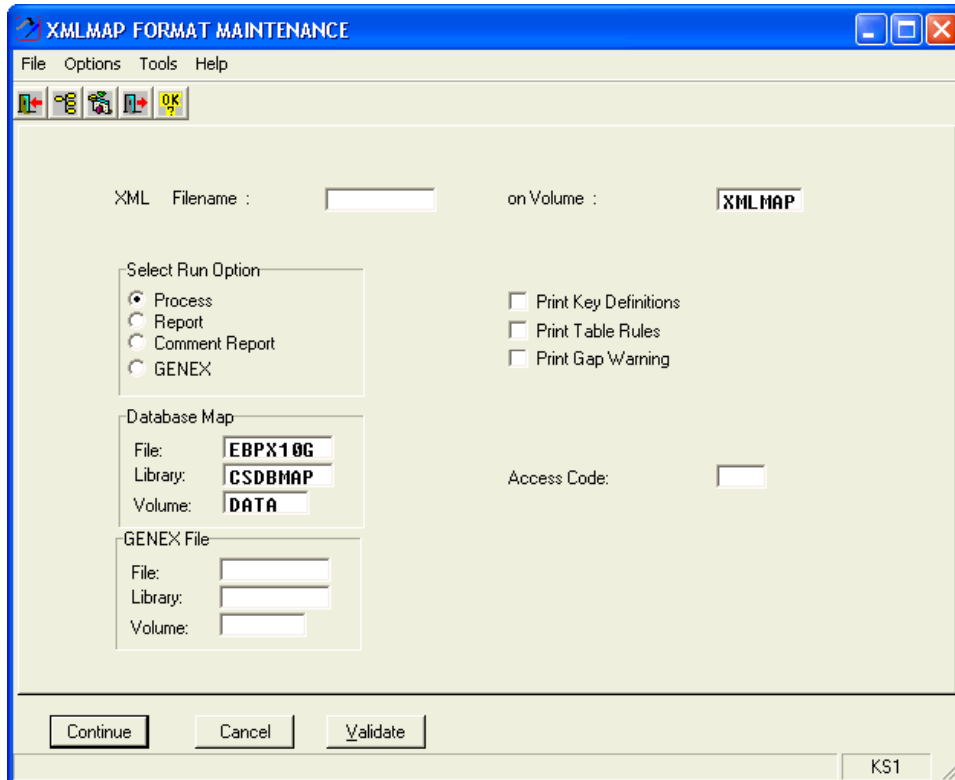
In order to use the XMLMaps, a specific entry must exist in the LGMAP file and the entry which points to the XMLMAP directory **must** be "XMLMAP".

Introduction to XMLMap Maintenance

Before maintaining XMLMaps, you must understand that there is a COPW and Data Description List (DDL) associated with each XMLMap. For each complete COPW/DDDL combination that Computron GENEX functions use there is a corresponding XMLMap. The COPWs are COBOL file layouts describing the required structure of the GENEX input file. Information such as file length, field lengths, data types, valid values and required record types is contained in the COPW. The COPWs used by each Computron GENEX function are described in detail in the corresponding application's Technical Guide. The XMLMap is comparable to the DDL and is required in order to GENEX-in a file that is in .xml format. If client-initiated changes are made to the user interface area of the DDL, the corresponding XMLMap must be modified to reflect the changes.

XMLMAP Format Maintenance Window (KS1)

The XMLMAP Format Maintenance Window (KS1) identifies the XMLMap that you want to view, print or maintain.



XMLMAP Format Maintenance Window (KS1)

XML Filename

Enter the name of the XMLMap to view, print or maintain. The XMLMap is the mechanism by which the system identifies the data in the .xml document to import.

on Volume

This is the location of the XMLMaps. This volume must exist in the LGMAP.

Default: XMLMAP, which is set by the system.

Select Run Option

Select the operation to perform. Valid options are:

Process – This option enables you to view the requested XMLMap online and modify the different records connected to this map.

Report – This option produces a printed report that contains information about the XMLMap. Information such as XML tags and record IDs are included on the report.

Comment Report – This option is currently not available.

GENEX – This option allows you to GENEX (Generalized External Interface) or import an XMLMap into the Comptron system. Enter the name of the new XMLMap in the XML

Filename field. This option is used with the GENEX File: File, Library and Volume fields, defined later.

Default: Process

Print Key Definitions

This field does not apply to XML maps

Print Table Rules

This field does not apply to XML maps

Print Gap Warning

This field does not apply to XML maps

Database Map File: / Library: / Volume:

Enter the name and location of the database map that determines the location of this XMLMap.

Access Code

This field is currently not implemented.

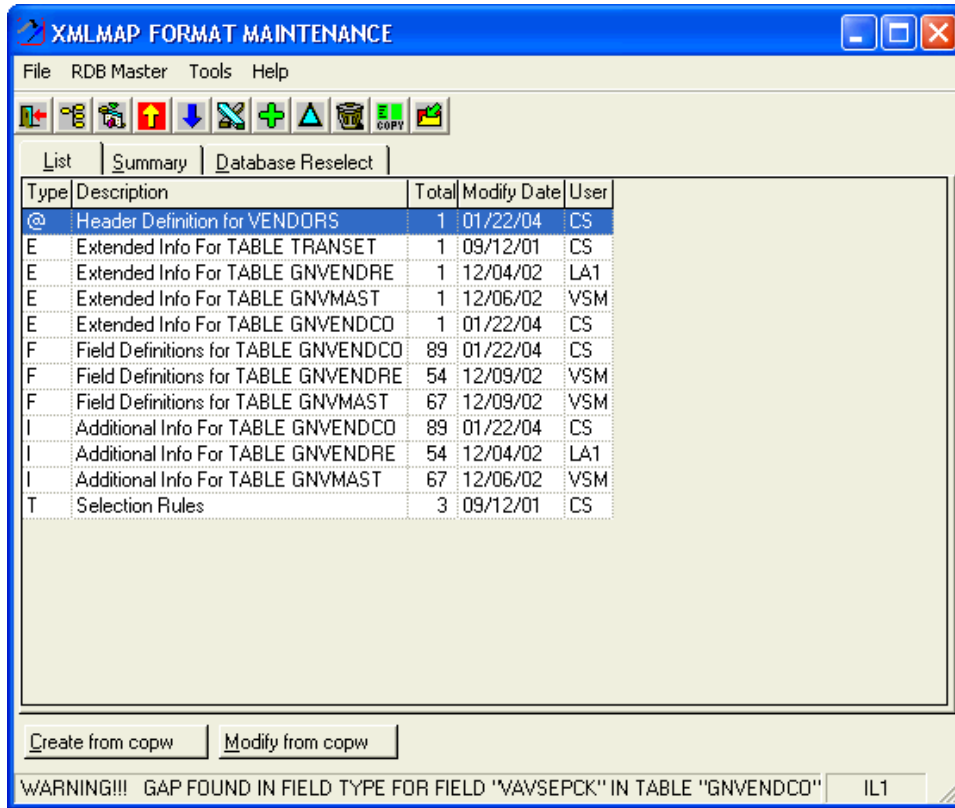
GENEX File: / Library: / Volume:

Enter the name and location of the input file to create an XMLMap. This is used when you select the GENEX option of the Select Run Option field. The file can be in either XML or flat file format. The format of the file is defined by the RDBGEX COPW and the rdbgenex DDL or the rdbgenex XML map. The file layout provides for the following record types:

Record Type	Description
P	Parameter record
T	Table record
K	Key information record
F	Field information record
E	Extended record

XMLMAP Format Maintenance Window (IL1)

To open the IL1 inquiry/list window, select the Process option in the Select Run Option field and click Continue.



The screenshot shows a window titled "XMLMAP FORMAT MAINTENANCE" with a menu bar (File, RDB Master, Tools, Help) and a toolbar. Below the toolbar are tabs for "List", "Summary", and "Database Reselect". The "List" tab is active, displaying a table with the following data:

Type	Description	Total	Modify Date	User
@	Header Definition for VENDORS	1	01/22/04	CS
E	Extended Info For TABLE TRANSET	1	09/12/01	CS
E	Extended Info For TABLE GNVENDRE	1	12/04/02	LA1
E	Extended Info For TABLE GNVMAST	1	12/06/02	VSM
E	Extended Info For TABLE GNVENDCO	1	01/22/04	CS
F	Field Definitions for TABLE GNVENDCO	89	01/22/04	CS
F	Field Definitions for TABLE GNVENDRE	54	12/09/02	VSM
F	Field Definitions for TABLE GNVMAST	67	12/09/02	VSM
I	Additional Info For TABLE GNVENDCO	89	01/22/04	CS
I	Additional Info For TABLE GNVENDRE	54	12/04/02	LA1
I	Additional Info For TABLE GNVMAST	67	12/06/02	VSM
T	Selection Rules	3	09/12/01	CS

At the bottom of the window, there are buttons for "Create from copw" and "Modify from copw", and a status bar displaying the warning: "WARNING!!! GAP FOUND IN FIELD TYPE FOR FIELD 'VAVSEPC' IN TABLE 'GNVENDCO'" and the window ID "IL1".

XMLMAP Format Maintenance Inquiry/List Window (IL1)

The system displays the file name and volume on the Summary tab based on the values in the XML Filename, on Volume and Database Map fields on the KS1 window. These fields are non-modifiable; however, it is important to verify that the XMLMap listed is the ones that you want.

Note: Computron's XMLMaps should not be modified except the User Interface Area.

Type

There are four types of records associated with each XMLMap. Valid entries are:

@ – Header record. This identifies the Extended record that lists the elements in the XML document. Elements are equivalent to records in a flat file.









A – Alias records. These records are optional and translate a long tag name to a shorter version.




E – Extended records. These records, except for the TRANSET, link GENEX file names to table names used by the relational database. The TRANSET identifies the required records in the GENEX file.

F – Field records. These records contain field definitions that describe each XML tag. The XML tag corresponds to a field in the file updated or column in the database updated.

T – Table records. These records contain the link between the record type and the XML format. The record type identifies the type of data in the record and what format is used.

I – These records link the XML tag to the COBOL code name. These are for information purposes only.

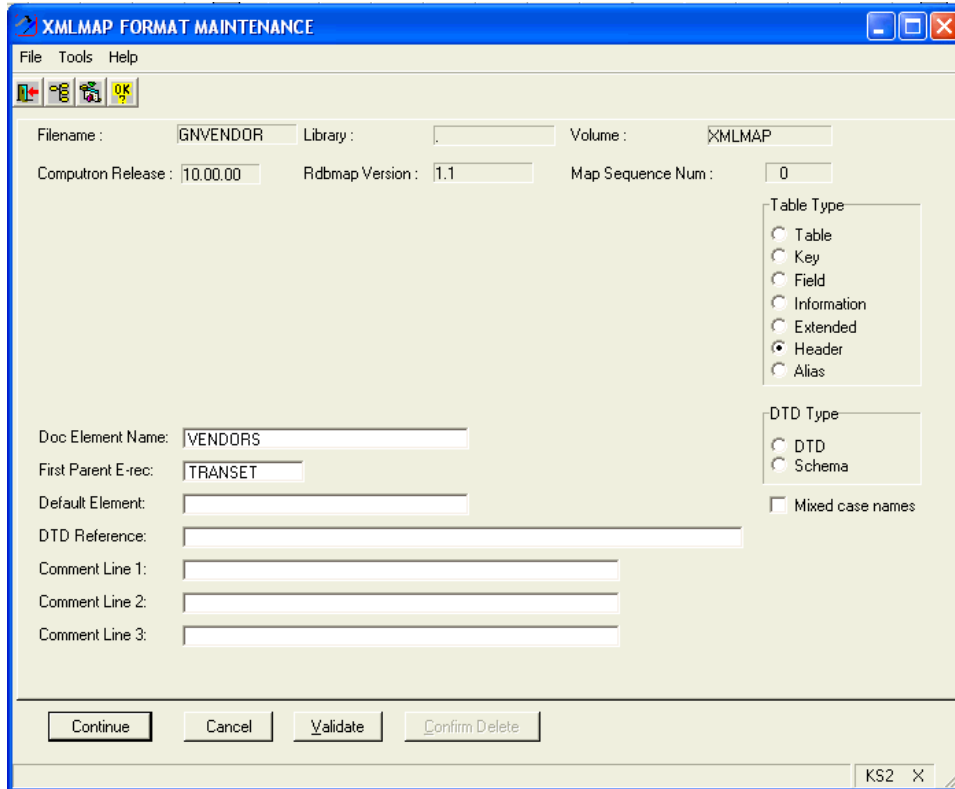
Options on the XMLMAP Format Maintenance Inquiry/List Window	
Button	Description
	Exit – This option exits the IL1 window of XMLMAP Format Maintenance.
	Previous – Returns to the previous window without exiting the XMLMAP function.
	More Data – Loads additional records in the UL1 window.
	Create new records for the current XMLMap.
	Modify existing records.
	Remove existing records.
	Create new records by duplicating an existing record and modifying it to form a unique record.
	Save – Fully implements the modifications to the XMLMap. Click this button after performing <i>any</i> of the other functions to save changes to the XMLMap.

Options on the XMLMAP Format Maintenance Inquiry/List Window	
Button	Description
	<p>Create new Field, Info and Extend records for the XMLMap based on the file layout directories contained in the current Computron software environment (CIXXCOPW and CIXXCOPY [where CIXX represents the system prefix entered on the XMLMAP Format Maintenance Window (KS1)]).</p> <p> <i>Caution! This option replaces any existing records. Make a backup of the existing XMLMap prior to completing this function.</i></p> <p><i>Validations:</i> The following validations are performed upon selection of this function:</p> <ul style="list-style-type: none"> ◆ This function verifies that the field type records account for all bytes in the file record and no gaps exist between the end of a defined field and the beginning of the next field. Any discontinuity results in a Gap Warning message on the window. ◆ Defined fields are ordered by starting position in the row. If fields are sequenced out of order, an error message is displayed and the user must render the Modify from copw option to resort the fields.
	<p>Modify existing Field, Info and Extended records or creates new records where appropriate, for the XMLMap based on the file layout directories contained in the current Computron software environment.</p> <p><i>Validation:</i> This function verifies that the field type records account for all bytes in the file record and no gaps exist between the end of a defined field and the beginning of the next field. Any discontinuity results in a Gap Warning message on the window.</p>

The headings that follow describe each record type in detail. The fields described for each record type are only the fields that apply to that record type.

@ (Header) Record

The @ (Header) record points to the Extended record containing the list of formats that make up the GENEX file. Usually this record points to TRANSET. There is only one of these for the XMLMap.



The screenshot shows the 'XMLMAP FORMAT MAINTENANCE' window with the following fields and options:

- Filename: GNVENDOR
- Library: .
- Volume: XMLMAP
- Computron Release: 10.00.00
- Rdbmap Version: 1.1
- Map Sequence Num: 0
- Table Type:
 - Table
 - Key
 - Field
 - Information
 - Extended
 - Header
 - Alias
- DTD Type:
 - DTD
 - Schema
- Mixed case names
- Doc Element Name: VENDORS
- First Parent E-rec: TRANSET
- Default Element: (empty)
- DTD Reference: (empty)
- Comment Line 1: (empty)
- Comment Line 2: (empty)
- Comment Line 3: (empty)

Buttons at the bottom: Continue, Cancel, Validate, Confirm Delete.

XMLMAP Format Maintenance Window (KS2) – @ Record

Table Type

For an @ record, this is set to Header.

Doc Element Name

This is the name of the GENEX file this XMLMap describes.

First Parent E-rec

This is the name of the Extended record that contains the list of record types for this GENEX file. The value in this field is generally TRANSET.

Comment Line 1, Comment Line 2, Comment Line 3

These three fields contain descriptive information about this XMLMap.

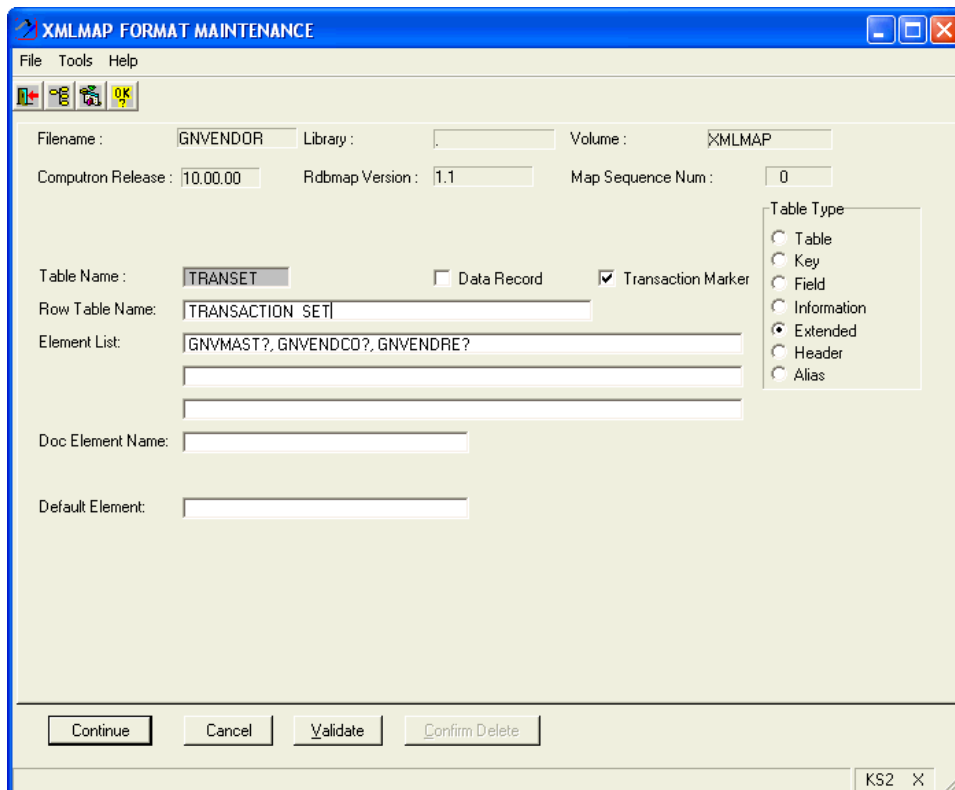
Extended Records

The Extended record contains the list of formats that make up a particular type of GENEX file entry (TRANSET, etc.) or a description of the format.

There are two types of Extended Records:

- ◆ The first type of Extended record is the TRANSET. It contains the list of records that make up the whole GENEX file. Required formats are followed by a question mark (?).

For example, for GNVENDOR or BATCHCST, etc., the formats in the Element List field are all or mostly required (AND). For GNVOUCHR or BATCHITM, etc., the TRANSET describes the types of transactions that are GENEXed into the system. These formats are OR included. There are further Extended records that describe what formats make up these transaction types.

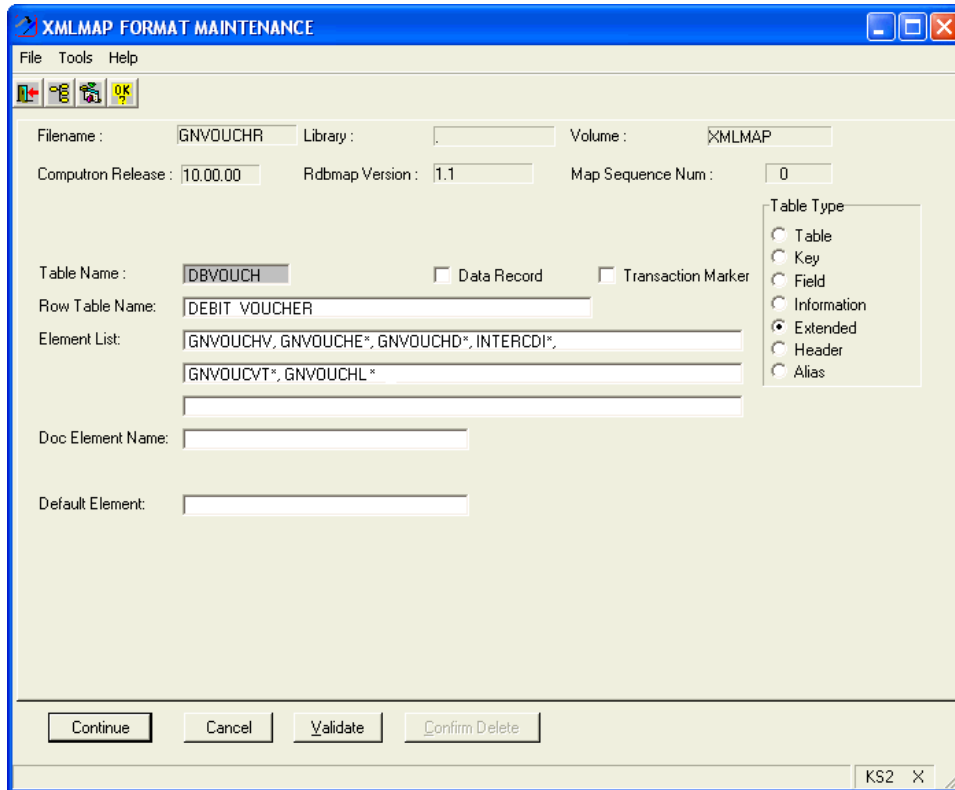


The screenshot shows the 'XMLMAP FORMAT MAINTENANCE' window. The 'Table Name' field is set to 'TRANSET'. The 'Table Type' dropdown menu is open, showing options: Table, Key, Field, Information, Extended (selected), Header, and Alias. The 'Element List' field contains 'GNVMAST?, GNVENDCO?, GNVENDRE?'. Other fields include 'Filename: GNVENDOR', 'Volume: XMLMAP', 'Map Sequence Num: 0', and 'Transaction Marker' checked.

XMLMAP Format Maintenance Window (KS2 X) – TRANSET Extended Record

- ◆ The second type of Extended record contains the list of formats that makeup one transaction. The optional record types are followed by an asterisk (*).

For example, the GNVENDOR or BATCHCST E records, other than the TRANSET, identify the multiple types of records needed to create a vendor or customer. The GNSOUCHR or BATCHITM E records, other than the TRANSET, identify formats that comprise one transaction or identify the types of records needed to create one transaction.



XMLMAP Format Maintenance Window (KS2) – Extended Records

Table Name

This is the eight character XMLMap name.

Table Type

For an E record, this is Extended.

Row Table Name

This is the description of what type of GENEX input this is. For example, for the E record for DBVOUCH or ARITEM, this is the type of transaction. For GNVOUCHV or BATCH1, this is the description of the type of record.

Element List

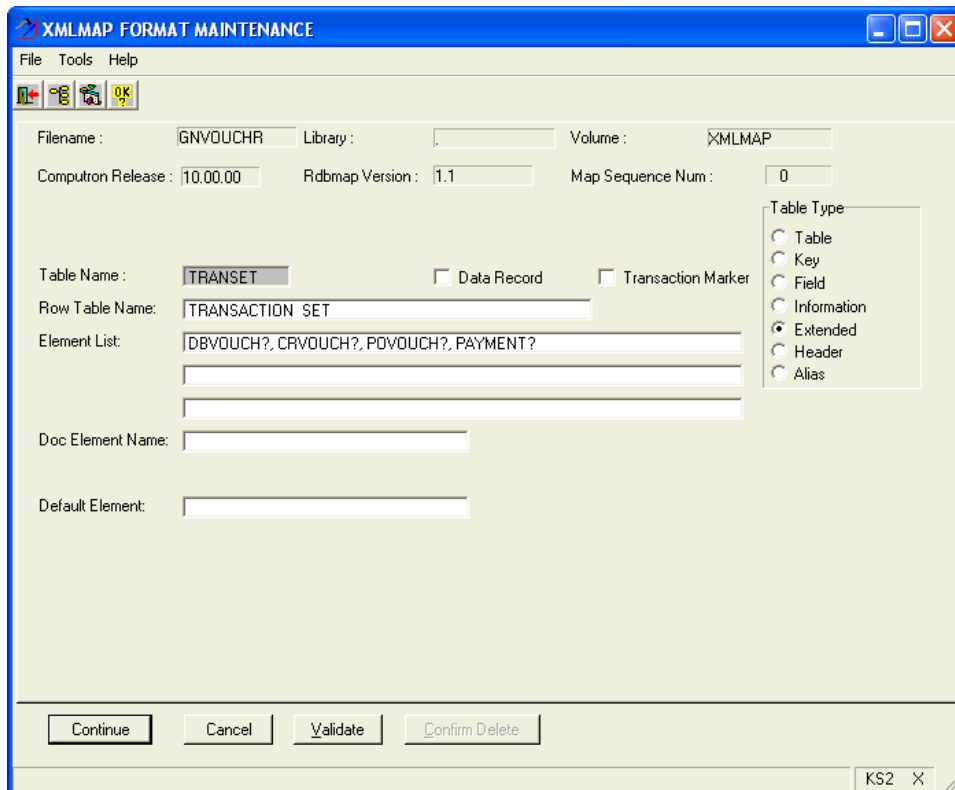
This field is only used when the E record describes a group of formats that comprise a complete GENEX input (usually multiple records with various formats). This is the list of record formats that make up this type of GENEX input. Because not all the records (fields of data) of an input are used at every client installation, GENEX files are broken down into

multiple records each containing some of the data for a complete GENEX input. Sometimes, the difference between one format and another is that they are different records for different files, e.g., the GENEX file for vendors, assets, customer, etc. Sometimes the difference between one format and another is that some of the data is optional and therefore in a different record, e.g., the GENEX file for vouchers, items, GL transactions, etc.

TRANSET Extended Record

TRANSET is a special E record that identifies the list of formats needed to complete the GENEX file. Sometimes the TRANSET points to another Extended record which contains this list, e.g., the BATCHITM TRANSET record points to the ARITEM E record.

In the example of GNSVOUCHR, the TRANSET Element List field includes DBVOUCH?, CRVOUCH?, POVOUCH?, PAYMENT?. Each of these is a transaction type, debit voucher, credit voucher, PO voucher or payment. You only need one of these transaction types to have a complete transaction. When you look at the DBVOUCH Extended record, the Element List field includes the formats GNSVOUCHV, GNSVOUCHE*, GNSVOUCHD*, INTERCDI*, GNSVOUCVT*, GNSVOUCHL*. All except the first one is optional.



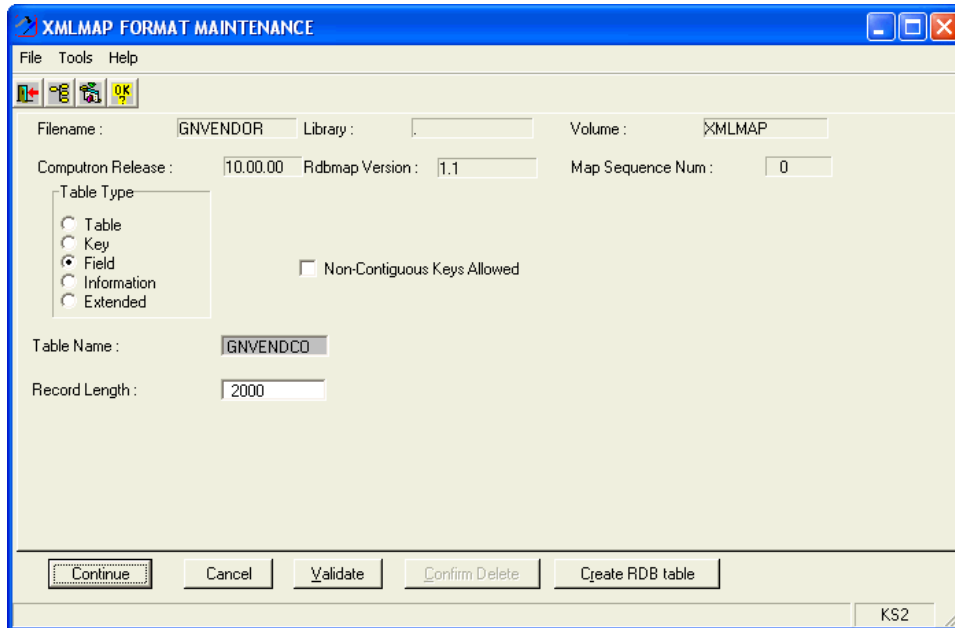
The screenshot shows the 'XMLMAP FORMAT MAINTENANCE' window with the following fields and options:

- Filename: GNSVOUCHR
- Library: (empty)
- Volume: XMLMAP
- Comptron Release: 10.00.00
- Rdbmap Version: 1.1
- Map Sequence Num: 0
- Table Name: TRANSET
- Data Record
- Transaction Marker
- Row Table Name: TRANSACTION SET
- Element List: DBVOUCH?, CRVOUCH?, POVOUCH?, PAYMENT?
- Doc Element Name: (empty)
- Default Element: (empty)
- Table Type: Extended
- Other Table Type options: Table, Key, Field, Information, Header, Alias
- Buttons: Continue, Cancel, Validate, Confirm Delete
- Status bar: KS2 X

XMLMAP Format Maintenance Window (KS2) – TRANSET Extended Record

Field Records

Field records identify the XML tags and their relationship to the DDL fields.



XMLMAP Format Maintenance Window (KS2) – Field Records

Table Type

For F records, this is Field.

Table Name

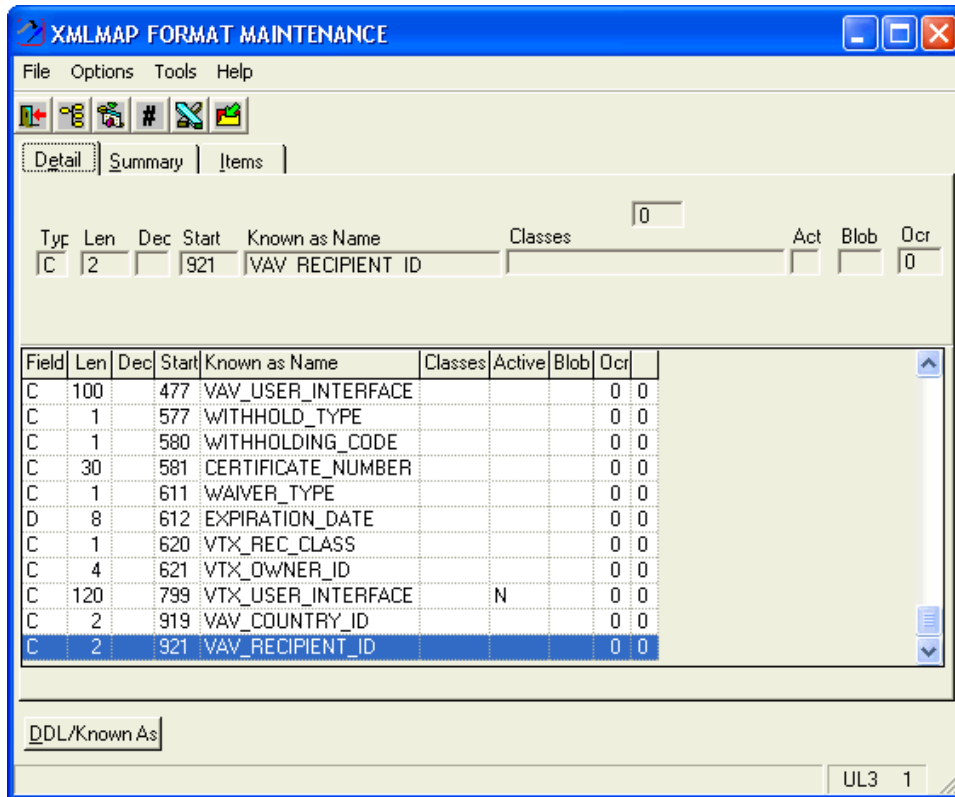
This is the eight character name of the GENEX format for this XMLMap.

Record Length

This is the total number of bytes in each record of this file, excluding tags.

- ☞ *Note:* This field may be extracted from the COPW file layout of the corresponding logical filename via the `Create from copw` function available from the XMLMAP Format Maintenance Inquiry/List Window (IL1).

The format details of each field are displayed on the UL3 window.



XMLMAP Format Maintenance Window (UL3) – Field Records

Field

This is the type of the data. For example, C = character, D = date, U = unsigned numeric, etc. Packed fields are not allowed in a GENEX file.

Len

This indicates the number of bytes this field occupies.

Dec

For numeric and decimal data types, this is the number of digits accepted to the right of the decimal point.

Start

This is the start position of the field in the input record.

Known as Name

This is the XML tag for this field.

Classes

This field is currently not implemented.

Active

This indicates the status of this field. Valid entries are:

Y/Blank – This indicates that the field is in use.

N – This indicates that the field is not in use. Typically, the user interface area is released as inactive. If your installation uses the user interface area to GENEX in proprietary data, this setting must be changed and the DDL and XML map must also be changed.

Blob

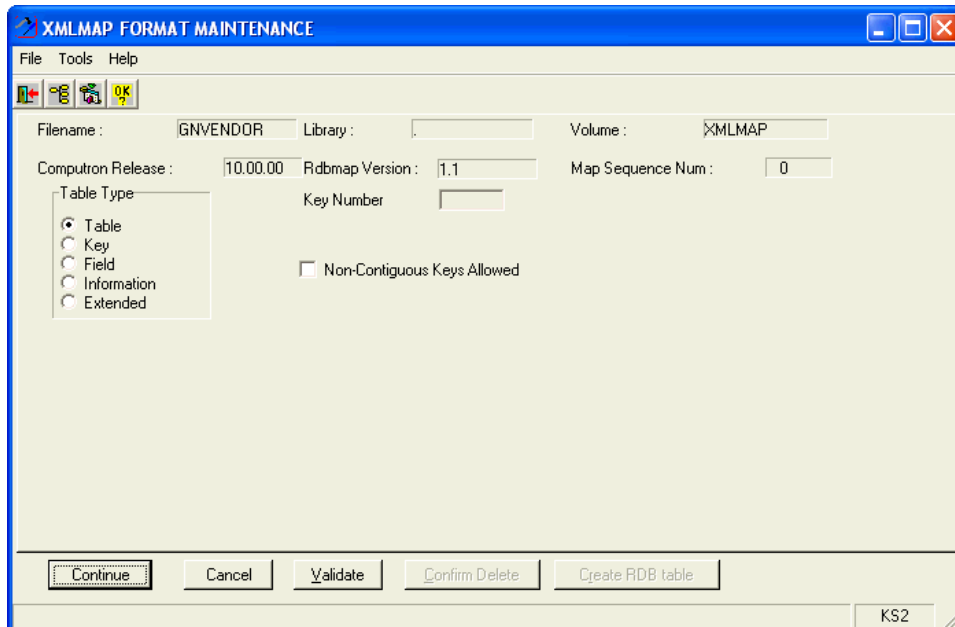
This is not used for GENEX files.

Occ

This is an “occurs” value. When multiple fields of data are formatted the same and contain basically the same data, all the fields could be described by one field occurring a number of times (occurs array). This feature is rarely used in GENEX formats.

Table Rules

Table rules are used by the system to tie the record type to the record format.

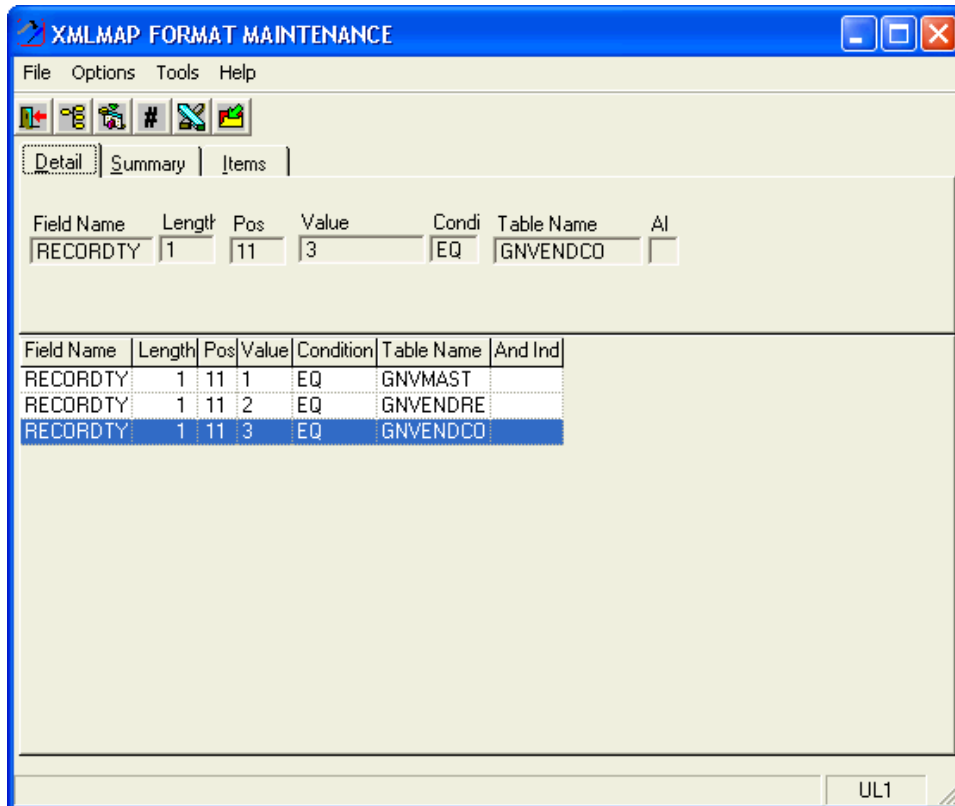


XMLMAP Format Maintenance Window (KS2) – Table Records

Table Type

For the T records, this is Table.

The rules that the application follows in order to determine the table required are defined on the UL1 window. For each record type (Value) there is a corresponding Table Name (format). The Field Name identifies which field contains the Value.



XMLMAP Format Maintenance Window (UL1) – Table Records

Field Name

This is the Data Description List (DDL) name of the field containing the value used as the record type.

Length

This is the number of bytes of the record type field.

Pos

This is the position in the record where the value can be found.

Value

This is the value of the record type to equate to the format.

Condition

This is the operator used for comparing the value in the rule with the value passed by the XML document. Valid values are:

EQ – This operator represents the rule, “Value must equal value passed by the XML document.”

NE – This operator represents the rule, “Value indicated in the rule is not equal to the value passed by the XML document.”

Table Name

This is the name of the format that corresponds to this record type.

And Ind

This field is currently not implemented.

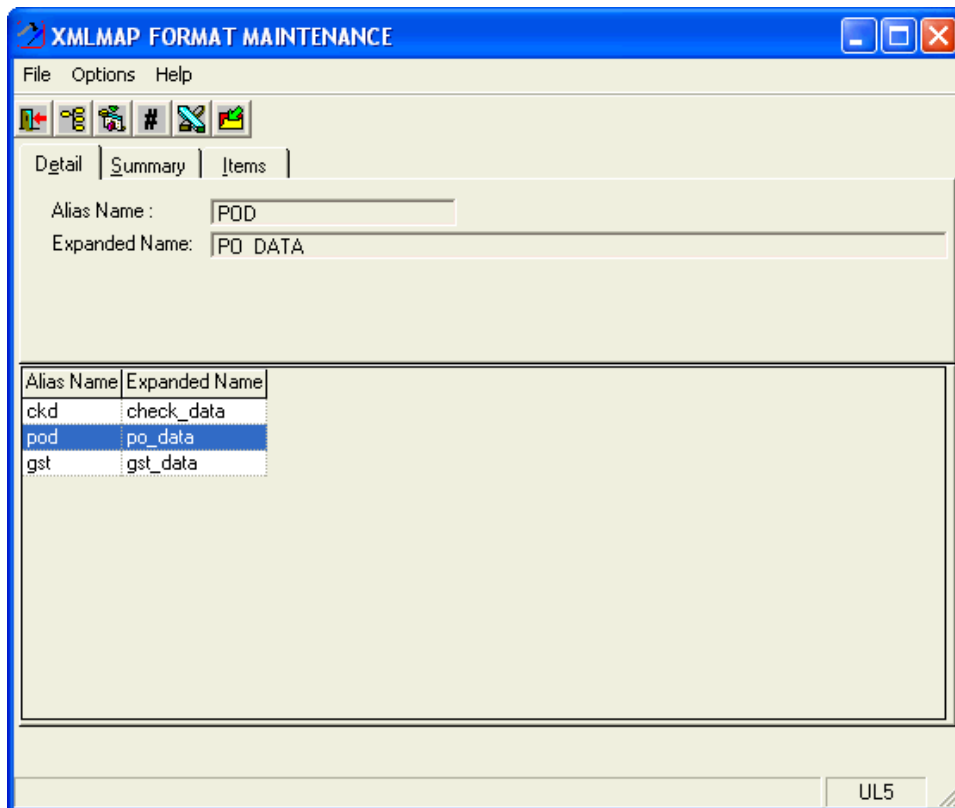
Info Records

Although Information records are displayed in the XMLMap, GENEX functions do not use this information.

Alias Records

Alias records are used for two purposes. This type of record is in a limited number of XMLMaps.

- ◆ To assign a name to a field that’s name is longer than the allowed 18 characters of “known-as” names.
- ◆ To simplify the naming of group items, e.g., address is a group of street, city, state and zip.



XMLMap Format Maintenance Window (IL1) – Alias Record

Report Option

A sample XMLMAP Format report is shown here:

```
<!ELEMENT vendors (transaction_set )>
<!ELEMENT transaction_set (ap_vendor?, vendor_company?, vendor_remit?)>
<!ELEMENT vendor_remit (vendor_id?, record_type?, remit_id?,
company_id?, sequence_id?, vendor_id_v1?, remit_id_v1?, active_status_ind?,
security_level?, address_line1?, address_line2?, address_line3?, city_name?,
state_id?, zip_code?, country_id?, telephone_number?, contact_name?,
reference_1?, other_text?, text_line1?, text_line2?, text_line3?,
record_class?, owner_id?, user_interface?, add_change_ind?, model_key?,
proc_ind?, external_hold_ind?, alternate_name1?, alternate_name2?,
eft_vendor_ind?, default_ccy_id?, vat_id?, vat_acct_id?, vat_id2?,
vat_acct_id2?, ap_acct_id?, cash_acct_id?, contact_name2?, tax_id?,
vat_origin_ind?, home_country_id?, interchg_id_qual?, interchg_id?,
org_id_qual?, org_id?, internal_reference?, exclusive_comp_id?, fax_number?,
telephone_extn?, govt_registration?, email_address?)>
<!ELEMENT ap_vendor (vendor_id?, record_type?, remit_id?,
company_id?, sequence_id?, vendor_id_v1?, active_status_ind?,
security_level?, sort_name?, soft_key?, soft_key2?, name_line1?,
name_line2?, reference_1?, hold_pmt_ind?, hold_pmt_amt?, group_id?,
employee_id?, one_time_vend_ind?, ap_1099_id?, ap_1099_form2_id?,
customer_id?, duns_number?, check_dup_vou_ind?, check_dup_inv_ind?,
check_dup_soft_key?, vat_cash_accr_ind?, chk_cr_bal_b4_pmt?,
check_corp_bal_ind?, num_invc_per_chk?, record_class?, owner_id?, tax_id?,
user_interface?, last_invoice_date?, last_paid_date?, add_change_ind?,
model_key?, short_name?, proc_ind?, external_hold_ind?, ap_1099_name?,
bkup_withhold_pct?, ap_1099_2nd_tin?, ap_1099_2nd_tin_dt?,
ap_1099_name_ctl?, auto_voucher_ind?, po_required_ind?, on_hold_for_po_ind?,
collective_vendor?, sub_contractor_ind?, certificate_type?,
certificate_number?, certificate_date?, expire_date_ind?,
match_tolerance_id?, update_catalog_ind?, ap_1042_country_id?,
ap_1042_recip_code?, home_country_id?, vat_origin_ind?, govt_registration?,
test_for_held_inv?, rise_security_lvl?, ap_1099_remit_id?, paid_on_po_ind?,
deduct_whld_ind?)>
<!ELEMENT vendor_company (vendor_id?, record_type?, remit_id?,
company_id?, sequence_id?, vendor_id_v1?, company_id_v1?, company_id_v2?,
factor_id?, active_status_ind?, security_level?, ap_acct_id?, cash_acct_id?,
separate_check_ind?, terms_id?, remit_id_v1?, default_ccy_id?,
payment_method_id?, category_id?, sub_category_id?, float_days?,
standard_msg?, standard_msg2?, std_distrib_id?, vat_id?, vat_acct_id?,
vat_id2?, vat_acct_id2?, check_sort_group?, record_class?, owner_id?,
user_interface?, last_invoice_date?, last_paid_date?, add_change_ind?,
model_key?, std_distrib_ind?, proc_ind?, trade_acct_id?, ship_via_id?,
ship_terms_id?, buy_from_id?, incr_gain_acct_id?, incr_loss_acct_id?,
antic_acct_id?, cr_note_net_ind?, whld_id?, whld_acct_id?, whld_id2?,
whld_acct_id2?, vav_sep_check_id?, payment_ccy_id?, vav_pmt_mthd_id?,
vav_cat_id?, vav_sub_cat_id?, vav_vat_id?, vav_vat_acct_id?, vav_vat_id2?,
vav_vat_acct_id2?, ap_1099_id?, ap_1099_form2_id?, hold_pmt_ind?,
hold_reason_id?, misc_reference_1?, misc_reference_2?, misc_reference_3?,
misc_reference_4?, vav_ap_acct_id?, vav_cash_acct_id?, discount_acct_id?,
antic_disc_acct_id?, trade_disc_acct_id?, hold_back_acct_id?,
dr_po_type_ind?, cr_po_type_ind?, batch_id?, vav_rec_class?, vav_owner_id?,
vav_user_interface?, withhold_type?, withholding_code?, certificate_number?,
waiver_type?, expiration_date?, vtx_rec_class?, vtx_owner_id?,
vtx_user_interface?, vav_country_id?, vav_recipient_id?)>
<!ELEMENT active_status_ind (#PCDATA)>
<!ELEMENT address_line1 (#PCDATA)>
<!ELEMENT address_line2 (#PCDATA)>
<!ELEMENT address_line3 (#PCDATA)>
<!ELEMENT add_change_ind (#PCDATA)>
<!ELEMENT alternate_name1 (#PCDATA)>
<!ELEMENT alternate_name2 (#PCDATA)>
<!ELEMENT antic_acct_id (#PCDATA)>
<!ELEMENT antic_disc_acct_id (#PCDATA)>
<!ELEMENT ap_1042_country_id (#PCDATA)>
<!ELEMENT ap_1042_recip_code (#PCDATA)>
<!ELEMENT ap_1099_2nd_tin (#PCDATA)>
<!ELEMENT ap_1099_2nd_tin_dt (date)>
<!ELEMENT ap_1099_form2_id (#PCDATA)>
<!ELEMENT ap_1099_id (#PCDATA)>
<!ELEMENT ap_1099_name (#PCDATA)>
<!ELEMENT ap_1099_name_ctl (#PCDATA)>
<!ELEMENT ap_1099_remit_id (#PCDATA)>
<!ELEMENT ap_acct_id (#PCDATA)>
<!ELEMENT auto_voucher_ind (#PCDATA)>
<!ELEMENT batch_id (#PCDATA)>
<!ELEMENT bkup_withhold_pct (#PCDATA)>
<!ELEMENT buy_from_id (#PCDATA)>
<!ELEMENT cash_acct_id (#PCDATA)>
<!ELEMENT category_id (#PCDATA)>
<!ELEMENT certificate_date (date)>
<!ELEMENT certificate_number (#PCDATA)>
<!ELEMENT certificate_type (#PCDATA)>
<!ELEMENT check_corp_bal_ind (#PCDATA)>
<!ELEMENT check_dup_inv_ind (#PCDATA)>
<!ELEMENT check_dup_soft_key (#PCDATA)>
<!ELEMENT check_dup_vou_ind (#PCDATA)>
<!ELEMENT check_sort_group (#PCDATA)>
<!ELEMENT chk_cr_bal_b4_pmt (#PCDATA)>
<!ELEMENT city_name (#PCDATA)>
<!ELEMENT collective_vendor (#PCDATA)>
<!ELEMENT company_id (#PCDATA)>
<!ELEMENT company_id_v1 (#PCDATA)>
<!ELEMENT company_id_v2 (#PCDATA)>
<!ELEMENT contact_name (#PCDATA)>
<!ELEMENT contact_name2 (#PCDATA)>
<!ELEMENT country_id (#PCDATA)>
<!ELEMENT cr_note_net_ind (#PCDATA)>
<!ELEMENT cr_po_type_ind (#PCDATA)>
<!ELEMENT customer_id (#PCDATA)>
<!ELEMENT deduct_whld_ind (#PCDATA)>
<!ELEMENT default_ccy_id (#PCDATA)>
<!ELEMENT discount_acct_id (#PCDATA)>
<!ELEMENT dr_po_type_ind (#PCDATA)>
<!ELEMENT duns_number (#PCDATA)>
<!ELEMENT eft_vendor_ind (#PCDATA)>
<!ELEMENT email_address (#PCDATA)>
<!ELEMENT employee_id (#PCDATA)>
<!ELEMENT exclusive_comp_id (#PCDATA)>
<!ELEMENT expiration_date (date)>
<!ELEMENT expire_date_ind (#PCDATA)>
<!ELEMENT external_hold_ind (#PCDATA)>
```

```

<ELEMENT factor_id (#PCDATA)>
<ELEMENT fax_number (#PCDATA)>
<ELEMENT float_days (#PCDATA)>
<ELEMENT govt_registration (#PCDATA)>
<ELEMENT group_id (#PCDATA)>
<ELEMENT hold_back_acct_id (#PCDATA)>
<ELEMENT hold_pmt_amt (#PCDATA)>
<ELEMENT hold_pmt_ind (#PCDATA)>
<ELEMENT hold_reason_id (#PCDATA)>
<ELEMENT home_country_id (#PCDATA)>
<ELEMENT incr_gain_acct_id (#PCDATA)>
<ELEMENT incr_loss_acct_id (#PCDATA)>
<ELEMENT interchg_id (#PCDATA)>
<ELEMENT interchg_id_qual (#PCDATA)>
<ELEMENT internal_reference (#PCDATA)>
<ELEMENT last_invoice_date (date)>
<ELEMENT last_paid_date (date)>
<ELEMENT match_tolerance_id (#PCDATA)>
<ELEMENT misc_reference_1 (#PCDATA)>
<ELEMENT misc_reference_2 (#PCDATA)>
<ELEMENT misc_reference_3 (#PCDATA)>
<ELEMENT misc_reference_4 (#PCDATA)>
<ELEMENT model_key (#PCDATA)>
<ELEMENT name_line1 (#PCDATA)>
<ELEMENT name_line2 (#PCDATA)>
<ELEMENT num_invcs_per_chk (#PCDATA)>
<ELEMENT one_time_vend_ind (#PCDATA)>
<ELEMENT on_hold_for_po_ind (#PCDATA)>
<ELEMENT org_id (#PCDATA)>
<ELEMENT org_id_qual (#PCDATA)>
<ELEMENT other_text (#PCDATA)>
<ELEMENT owner_id (#PCDATA)>
<ELEMENT paid_on_po_ind (#PCDATA)>
<ELEMENT payment_ccy_id (#PCDATA)>
<ELEMENT payment_method_id (#PCDATA)>
<ELEMENT po_required_ind (#PCDATA)>
<ELEMENT proc_ind (#PCDATA)>
<ELEMENT record_class (#PCDATA)>
<ELEMENT record_type (#PCDATA)>
<ELEMENT reference_1 (#PCDATA)>
<ELEMENT remit_id (#PCDATA)>
<ELEMENT remit_id_v1 (#PCDATA)>
<ELEMENT rlse_security_lvl (#PCDATA)>
<ELEMENT security_level (#PCDATA)>
<ELEMENT separate_check_ind (#PCDATA)>
<ELEMENT sequence_id (#PCDATA)>
<ELEMENT ship_terms_id (#PCDATA)>
<ELEMENT ship_via_id (#PCDATA)>
<ELEMENT short_name (#PCDATA)>
<ELEMENT soft_key (#PCDATA)>
<ELEMENT soft_key2 (#PCDATA)>
<ELEMENT sort_name (#PCDATA)>
<ELEMENT standard_msg (#PCDATA)>
<ELEMENT standard_msg2 (#PCDATA)>
<ELEMENT state_id (#PCDATA)>
<ELEMENT std_distrib_id (#PCDATA)>
<ELEMENT std_distrib_ind (#PCDATA)>
<ELEMENT sub_category_id (#PCDATA)>
<ELEMENT sub_contractor_ind (#PCDATA)>
<ELEMENT tax_id (#PCDATA)>
<ELEMENT telephone_extn (#PCDATA)>
<ELEMENT telephone_number (#PCDATA)>
<ELEMENT terms_id (#PCDATA)>
<ELEMENT test_for_held_inv (#PCDATA)>
<ELEMENT text_line1 (#PCDATA)>
<ELEMENT text_line2 (#PCDATA)>
<ELEMENT text_line3 (#PCDATA)>
<ELEMENT trade_acct_id (#PCDATA)>
<ELEMENT trade_disc_acct_id (#PCDATA)>
<ELEMENT update_catalog_ind (#PCDATA)>
<ELEMENT user_interface (#PCDATA)>
<ELEMENT vat_acct_id (#PCDATA)>
<ELEMENT vat_acct_id2 (#PCDATA)>
<ELEMENT vat_cash_accr_ind (#PCDATA)>
<ELEMENT vat_id (#PCDATA)>
<ELEMENT vat_id2 (#PCDATA)>
<ELEMENT vat_origin_ind (#PCDATA)>
<ELEMENT vav_ap_acct_id (#PCDATA)>
<ELEMENT vav_cash_acct_id (#PCDATA)>
<ELEMENT vav_cat_id (#PCDATA)>
<ELEMENT vav_country_id (#PCDATA)>
<ELEMENT vav_owner_id (#PCDATA)>
<ELEMENT vav_pmt_mthd_id (#PCDATA)>
<ELEMENT vav_recipient_id (#PCDATA)>
<ELEMENT vav_rec_class (#PCDATA)>
<ELEMENT vav_sep_check_id (#PCDATA)>
<ELEMENT vav_sub_cat_id (#PCDATA)>
<ELEMENT vav_user_interface (#PCDATA)>
<ELEMENT vav_vat_acct_id (#PCDATA)>
<ELEMENT vav_vat_acct_id2 (#PCDATA)>
<ELEMENT vav_vat_id (#PCDATA)>
<ELEMENT vav_vat_id2 (#PCDATA)>
<ELEMENT vendor_id (#PCDATA)>
<ELEMENT vendor_id_v1 (#PCDATA)>
<ELEMENT vtx_owner_id (#PCDATA)>
<ELEMENT vtx_rec_class (#PCDATA)>
<ELEMENT vtx_user_interface (#PCDATA)>
<ELEMENT waiver_type (#PCDATA)>
<ELEMENT whld_acct_id (#PCDATA)>
<ELEMENT whld_acct_id2 (#PCDATA)>
<ELEMENT whld_id (#PCDATA)>
<ELEMENT whld_id2 (#PCDATA)>
<ELEMENT withholding_code (#PCDATA)>
<ELEMENT withhold_type (#PCDATA)>
<ELEMENT zip_code (#PCDATA)>
<ELEMENT date (month, day, year)>
<ELEMENT month (#PCDATA)>
<ELEMENT day (#PCDATA)>
<ELEMENT year (#PCDATA)>

```

6.7 UEXMAP Format Maintenance

Overview

The Computron UEXMAP files serve as external schema that defines the required structure of runtime User Exit files that are used in interfacing with custom applications. The records of the UEXMAP file are used to map the runtime to the custom interface program.

Please contact Computron for information regarding creating runtime UserExits.

6.8 Server Log Files

Introduction to Server Log Files

In the Computron Unix implementation, there are a number of different log files that are updated with information useful for tracking and analyzing the processing on the application server. This section identifies various log files and describes the type of information they contain and how and when they are updated.

User Log File (user_XXX)

For each user running the Computron application, a log file unique to that user's ID is created. For releases prior to 7.0, the user log files were by default located in the `/tmp` directory and the file naming convention was "`ctron_log_{userid}`". As of 7.0 the log files are contained in the directory `$CTRON_ROOT/var/logs` and the file naming convention is "`user_{userid}`". The file is owned by the user and is in the `ctron` group. It gets trimmed by the `cscron` process (Refer to Section 5.8 of the *System Administrators Guide for Unix* for more details concerning the `cscron` function.).

The user log file is written to by the runtime. Unless specific debugging options are set in the processing environment, only error messages are output to the log. If no debugging is enabled and there are no irregularities in the environment, within the applications or improper processing by the user (e.g. aborting out of a process abnormally, or exiting an AUI session improperly), the user's log file remains empty.

As mentioned above, there are various debugging options available to the user. These options cause different types of information to be written to the user's log file. (Refer to Chapter 6 of the *System Administrators Guide for Unix* for more details concerning debugging options.)

Background Processing Log File (cseqd.log)

As of release 7.0, the recommended implementation for background processing is the functionality known as `cseq`. Depending on the logging level enabled, information can be written to a log file specifically created for this purpose. The log is located in `$CTRON_ROOT/var/logs` and the name of the log file is `cseqd.log`. The file is owned by `ctronsys` and is in the `ctron` group. (Refer to Chapter 5.2 of the *System Administrators Guide for Unix* for more details concerning `cseq` and logging levels for `cseq`.)

CSSERVE Log File (csserve.logfile)

This log file contains information about the `csserve` processes running on the server and is written only when logging is enabled in the `csserv.conf` file, a file located in `$CTRON_EXEC/_ctron_`. The entry that enables logging and also identifies the file being logged to is:

```
LOGFILE={/tmp/csserve.logfile}
```

It logs connects and disconnects for the `csserve` processes in that environment, as well as information about when services get started and some other details. It is owned by root and as illustrated in the example of the `csserv.conf` entry, it is typically located in the `/tmp` directory.

Audit Log File (audit_log)

The `audit_log` file gets updated by two main processes:

<code>create_db</code>	All of the functions that are part of the RDBMS <code>create_db</code> utility write a record of the processing command, the results of the command, the date/time stamp, user ID, process ID, and any errors encountered to the <code>audit_log</code> file. The same information is also written to the user's log file.
<code>cshldclr</code>	When the holdlist is cleared, the <code>audit_log</code> file gets updated with the <code>cshldclr</code> command used, the date/time stamp, the user running the command, the user's process ID, and the key to the holdlist records that were cleared.

The `audit_log` file is located in `$CTRON_ROOT/exec/_ctron_`. The file is owned by `ctronsys` and is in the `ctron` group. This log is truncated by `cscron` to the number of lines specified in the line in `cscron` beginning "truncate_AUDITLOG", which is, when released, set to the value 50,000.

System Log File (sys_log)

As of the 7.0 release, only a few processes write to the `sys_log` file. These are the `truncate_tables` and `drop_tables` utilities but only when certain types of errors are encountered. The RDBMS runtime writes to this log if there is a problem deleting the runtime cache. The information written to this log file may be expanded in future releases.

Appendices

Appendix A: Printer Setup

Introduction to Printer Setup

Computron uses the printer control files, LPMAP, PRMAP and FORMS to direct output from the applications to the Unix `lp` print service. The files contain information that enables the system to translate the parameters accompanying a print request from a Computron application into Unix `lp` and `lpstat` commands. The control files are located in the `.../exec/_ctron_/wisp/config` directory (where `...` represents the directory name of the file system where the Computron software resides) and can be edited using a text editor such as `vi`. Each file is described in detail below.

☞ *Note:* Any printer that is intended for use with a Computron application must first be configured and operational within the Unix environment.

File Descriptions

LPMAP

The LPMAP file lists the user-defined print classes to be used by Computron applications. These print classes control the printer characteristics to set for the print job. The layout of the file is as follows:

- ◆ twenty six print classes can be defined in LPMAP;
- ◆ each line of the file defines one print class;
- ◆ print class entry (A–Z) is entered as the first character of each line;
- ◆ the second character of each line is always a space;
- ◆ the remainder of the line is the class specific printer option to be used.

Example:

```
A -o no banner
B -o draft
.
.
Z -d lp 0
```

PRMAP

The PRMAP file defines logical printers used by Computron applications. This file defines the printer numbers used by Computron applications. Printer numbers determine the `lp` print service device to which print files are routed. The file layout is as follows:

- ◆ up to 254 Printer Numbers can be defined in PRMAP;
- ◆ each line of the file defines one printer number;
- ◆ the first three characters of each line consist of the printer number entry (001–254);
- ◆ a space is entered as the fourth character of each line;
- ◆ the remainder of the line is the destination option (`-d`) to the `lp` command and printer as defined in the `lp` print service.

Example:

```
001 -dlp0
002 -dlp2
.
.
254 -dlp#
```

Forms

The FORMS file defines FORM_NUM codes that are used by the WISP Command Processor (accessible from Computron applications). FORM_NUM codes correspond to predefined printer options such as characters per inch, font, landscape, portrait, etc. The structure of the FORMS file is as follows:

- ◆ up to 254 printer FORM_NUM codes can be defined in the FORMS file;
- ◆ each line of the file defines one code;
- ◆ a numerical entry (001–254) representing the FORM_NUM code constitutes the first three characters of each line;
- ◆ the fourth character of each line is always a space;
- ◆ the remainder of the line is the option name defined in the `lp` print service, or options as they would be entered as part of an `lp` command at the Unix command prompt.

Example:

```
001 -ow132 -ol66
002 -oLANDSCAPE
.
.
254 -oPOSTSCRIPT
```

Interface to the Unix Print Spooler

After selecting the print class, printer number and form number, the Unix shell script `csprint.sh` executes. This shell script provides the interface to the Unix print spooler. Using the previous examples, if you select:

Printer Class	A
Printer Number	002
Form Number	001
Copies	3

the following Unix command executes:

```
lp -c -s -n3 -ow132 -ol66 -oNOBANNER -dlp2
```

If you select,

Printer Class	2
Printer Number	2
Form Number	001
Copies	3

the following Unix command executes:

```
lp -c -s -o3 -ow132 -ol66 -dlp0 -dlp2
```

Most Unix systems use the second `-d` specification in this case. It is generally a good practice to set up the LPMAP, PRMAP and FORMS entries to disallow such ambiguities.

Forms and Fonts

Computron's printer control files can be used to interface with a Forms and Fonts filter. Forms and Fonts is a print filter that adds enhanced printing capabilities to Unix systems. This software provides the ability to control the font and the margins or page setup used to print a file. Additionally, this software enables users to create and store overlays. (Overlays are electronic "forms" that can be used to simulate pre-printed documents such as letterhead, purchase orders, etc.) Forms and Fonts allows overlays to be used in conjunction with print files.

Implementation of the Forms and Fonts filter with the Computron applications requires the forms and fonts entries in the printer control files PRMAP and FORMS.

PRMAP

As explained earlier, the PRMAP file defines logical printers used by Computron applications. Each logical printer is assigned a printer number as well as the appropriate print command to use for the print service and the device. To designate a Forms and Fonts printer in this file, use the following format:

```
### !formfont "forms & fonts parameter list" "lp parameter list"
```

where `###` represents the printer number, "forms & fonts parameter list" represents any Forms and Fonts parameters and "lp parameter list" represents lp parameters for printer-specific options

Example:

To define the logical printer 025 (printer number 025 from the Computron applications) to interface with Forms and Fonts passing the parameters, `-pcl`, and to use the device, `laserprt` with the options, `-o nb`, create the following entry:

```
025 !formfont "-pcl" "-d laserprt -o nb"
```

Forms

The FORMS file defines FORM_NUM codes that are used by the WISP Command Processor (accessible from Computron applications). FORM_NUM codes correspond to predefined printer options, such as characters per inch, font, landscape, portrait, etc. Entries can be added to this file to define a form name to pass to Forms and Fonts when processing a particular print job. If a user enters a form number that does not exist in the Computron FORMS file, the form number is passed directly to Forms and Fonts for processing.

How the Interface Works

The Forms and Fonts printing functions are accessed via Computron's Print Queue by selecting the appropriate printer number and form number designation. The Computron print management mechanism reads the printer control files to set up the appropriate print commands. Specifically, the Unix shell scripts, `/. . . /exec/_ctron_/formfont.sh` and `/. . . /exec/_ctron_/csprint.sh`, read the files PRMAP and FORMS.

Appendix B: UserExits

Introduction to UserExits

All Computron systems have been developed in a way that allows the user to exercise the wide range of features and functions available in a parameterized environment, while maintaining the ability to interface with existing systems. One of the modules specifically designed to allow interaction between Computron's standard programming and non-Computron systems and subroutines is the UserExit. This appendix provides the user with a detailed description of UserExits, along with all the information needed to implement UserExits in the local processing environment.

To understand how UserExits are implemented, there are several basic questions that must be answered:

What Are UserExits?

UserExits allow users to interface non-Computron subroutines (programs) with certain Computron programs. This feature accommodates users who need to extend Computron's systems to perform additional validations or file updates, while maintaining standard processing within Computron's system.

For example, in the Accounts Payable system when entering the expense reports of employees, you may want to verify employee numbers against an external payroll or HR system. When paying an invoice, you may want to verify the terms of a contract by referring to a contract file.

How Do UserExits Work?

Selected Computron programs are able to call the UserExit subroutine to perform a custom function. Generally, the programs that include this capability fall into the categories of data entry and processing type functions. Inquiry/list and reporting functions do not include UserExits. Code maintenance functions, which are updated via the standard entry program CENTRY, incorporate the ability to use UserExits. However, UserExits are implemented in that program by a method different from that documented here (contact Computron for more information).

The UserExit subroutine can be written by Computron for an additional charge, although the client's own IT staff should be able to write it. A sample UserExit subroutine, XXUEXITS is provided for guidance with the release software. Each of the Computron financial applications has its own UserExit. The programs within each financial application are capable of calling the UserExit (XXUEXIT, where XX represents the application - GL, AP, AR, etc.) at selected points.

The UserExit subroutine can be written in such a way as to perform many different processes or to call other subroutines to perform these processes. The action that is performed depends on the Computron program and window from which the subroutine is called. For example, in the Accounts Payable system, APUEXIT could validate the purchase order number entered on one window of the Voucher Entry function (ICENTRY) and validate the posting account identifiers entered on a later window in the same function.

A call for the UserExit subroutine to perform a specific action is made at the following points in the Computron program:

- ◆ At the beginning of the program after Computron’s files are opened and the system’s Global Parameters file is read. This is referred to as the Start action.
- ◆ For ULM windows, a call to the UserExit is made prior to the display of the entry area for each new line item. This is the Before action. This action is commonly used to establish default values for fields that have been defined to allow entries into the User Interface Area.
- ◆ Once after each Computron program window or, for windows with multiple line items, once after each line item. Some windows, such as those prompting for company number, are excluded. The call to the UserExit is made after the window has been displayed and Computron’s standard validations have been performed for all fields on the window. This is referred to as the Validate action. No UserExit updates to data can be performed when called with the Validate action.

In addition, the Validate action can be optionally used to revalidate each ULM line just prior to Computron updates. The UserExit Revalidate Lines field on the Global Parameters Maintenance Window determines if this action is in use. Refer to your product’s Operations Guide for more information about this field.

- ◆ Another type of validation that can be performed once after a ULM window is the Final-Validation action. This allows the user to perform a final validation after the ULM window. This action is generally used for performing a validation on total fields which have been accumulated on the ULM window and is performed after the user selects the Process/Save option and before the Process action (see below).
- ◆ Once in each program or, in some cases, once per transaction after Computron’s standard file updates have been performed. This is referred to as the Process action. This is where the UserExit can perform its updates.
- ◆ At the end of the program before Computron’s files are closed. This is referred to as the Terminate action.

In summary, the UserExit subroutine can be called within each calling program at the following points:

Action	When Performed
Start	Beginning of program, before display of first window.
Before	Just before the display of a ULM line item.
Validate	Once after each program window or line item (for ULM windows). The revalidate action (if in use) is performed following the Computron validations which are performed when the user selects the Process/Save option from a ULM window.

Action	When Performed
Final-Val	For ULM windows only, just after the final Computron process, following the use of the Process/Save option which initiates file updating.
Process	Once in each program (e.g., after the user selects the Enter/Continue option at the last window) or after the user selects the Process/Save option which initiates file updating for ULM windows, and after Computron's standard file updates.
Terminate	At the end of each Computron program, before Computron files are closed.

The following are examples of how the UserExit can be used to extend or customize Computron programs:

Validations

After the information entered by a user on a window is accepted by Computron's validations, the data is passed to a hold area. The UserExit subroutine can perform additional validations on the information in the hold area before returning to Computron's program. In addition, the UserExit can return an error message to the originating Computron window if the data is invalid. To the operator, the error message appears to be a Computron validation. Computron's program verifies that the UserExit validation action has been completed before continuing to the next window. How the validation action is handled by the UserExit is totally dependent upon the user's needs and internal procedures. For example, the AP UserExit can:

- ◆ Validate the posting account identifiers entered in Computron's Accounts Payable system against an external general ledger system. If the entered posting account identifier does not match an account in the external system, an error message is returned to the operator on the originating window.
- ◆ Validate the employee numbers entered in a reference field in Computron's Accounts Payable system (when recording employee expenses) against an external payroll system.
- ◆ Validate entries to a custom data field which was added to the User Interface Area (UIA) of a standard Computron file and to a Computron window using Soft Technology feature.
- ◆ Capture additional information from external systems and return it to the UIA of a standard Computron file.
- ◆ Add new cross-validations between specified custom and/or Computron data fields.

Processing

Upon completion of Computron's processing (file updates), the UserExit subroutine can be called to perform additional file updates. In transaction maintenance programs, information is passed for the file being maintained from Computron's system to a hold area from which it can be extracted as needed.

Then, this information can be used to update a file in an external system. For example, the AP UserExit can:

- ◆ Update an external payroll system with employee expense information after a voucher is processed (created) in Computron's Accounts Payable system.
- ◆ Update an external purchase order system with the voucher number assigned after a voucher is processed.

☞ *Note:* The UserExit can display additional windows, allowing the user to enter information in addition to that prompted for on Computron's windows. For example, instead of using a reference field in Computron's system to enter employee numbers, the user can utilize a field appearing on a window displayed by the UserExit.

Where Are UserExits Implemented?

The User Exit interface has been implemented for the following Computron systems: Universal Utilities, Accounts Payable, Accounts Receivable, Fixed Assets, General Ledger, Inventory, Purchasing, and TEAM. The following is a list of the Universal Utility and Accounts Payable Computron programs in which the UserExit interface has been implemented.

Universal Utilities

Calendar Year Maintenance	CALYRMNT
Batch Stream User Code Maintenance	BSUSERM2
Alternate Key Maintenance (as implemented within AP, AR, and FA systems)	DIMSETMT

Accounts Payable

Company Parameters Maintenance	APCOPMNT
Company Accounts Maintenance	APCOAMNT
Bank Account Maintenance	APBNKMNT
Payment Terms Maintenance	APTERMNT
Vendor Maintenance	VMAINT
Batch Control Maintenance	APBATMT0
Voucher Entry	ICENTRY
Voucher Change	ICCHANG
Voucher Transfer/Reverse/Delete	ICTRANS
Voucher Hold/Release Maintenance	VOUHRMNT
Recurring Voucher Maintenance	RVOUMNT

Recurring Voucher Process	RVOUPST0
Miscellaneous Entry/Edit	RGENTRY
Pay Vouchers/Write Checks	WRITECKF
Manual and Damaged Checks	MCVENTRY
Void Check Entry	VOIDCHCK
Check Reconciliation and Bank Transactions	CKRENTRY
Posting	APREGPPR and APREGPP0
End Of Period / End of Year	APEOP0
GENEX-In Vendors	VENDGEN0
GENEX-In Vouchers	ICGEN0
GENEX-In Check Reconciliation	TAPETOCK
GENEX-In Factors	FGEN0
Vendor Statistics Maintenance	VSTATMNT

Implementation Overview

This section describes how to implement the UserExit in the Computron Accounts Payable system. However, the process is identical for each Computron application system — simply replace “AP” with the appropriate system prefix (AR, FA, GL, IV, PO, TB, or UV). The implementation of a UserExit for each supported environment is described below:

Unix Implementation Using ACUCOBOL

The implementation of a UserExit with ACUCOBOL’s version 4.1 compiler is a seven-step process:

1. Write the UserExit subroutine. Use the shell UserExit subroutine described under the heading XXUEXITS later in this Appendix as a model. Note that the name for the subroutine must be XXUEXIT, where XX represents the application for which the UserExit is being written (AP, AR, FA, GL, PO, TB, etc.).
2. Create a directory to designate the location of the object file. It is imperative that this directory be specified as `ybxload`, where `xx` is a meaningful designation for your company.
3. Create a directory to designate the location of an error log. If any errors occur during the COBOL compilation of the UserExit, a file is created in this directory listing the errors.
4. Compile the UserExit using the “ccbl” command as follows:

```
ccbl -w -sa -ce=COP -Zl -Rw ESCAPE -Da4 -Dz -Zd -o objfilepath -e errfilepath  
srcfilepath
```

where: `-ce=COP` is applicable if copy members are identified with a `.COP` extension,
`-Rw` is to remove “CRT” and “ESCAPE” from the COBOL reserved word list,
`-o objfilepath` is to designate the path of the object file, e.g.,
`/ctron/exec/ybxxload/APUEXIT`,
`-e errfilepath` is to designate the path of an error log, e.g.,
`/ctronsrc/ctl/ybxx/APUEXIT.ERR`,
`srcfilepath` is to designate the location of the source file e.g.,
`/ctronsrc/src/src/ybxx/APUEXIT.COB`

The remaining arguments are needed for the COBOL compilation designations (e.g., word alignment), and the above compile conventions should be used.

5. Establish path for access to the compiled UserExit. In the Unix environment, executable programs and subroutines are located through a path of directories as defined in Computron’s configuration parameter, `CODE_PREFIX`. Custom file handling is set up as follows:

- a) Create a file called `.custom` in `/ctron/exec/_ctron_`.
- b) Edit the `.custom` file. The file needs to contain exactly four letters on one line. For example, `ybxx`, where these four letters match the designation of the load directory.

This would cause `/ctron/exec/ybxxload` to be searched before any other load directory. This `ybxxload` directory is where the custom UserExit must reside. (See the `-o` argument in the `ccbl` command, above.)

When the application is invoked using the standard Computron start script, the `.custom` file causes the configuration parameter, `CODE_PREFIX`, to be set accordingly.

6. Modify the Global Parameters. To modify the Accounts Payable parameters, run the Global Parameters Maintenance function (GLOBAPMT). Entries are required in the following fields:

UserExit Validations

This field determines whether the UserExit Validation action is performed, and, if so, whether the CRT (or window) remains open. Valid options are:

No – The UserExit Validation action is not in use.

Yes (Don’t close CRT) – The UserExit Validation action is in use. In addition, the CRT (or window) stays open in the main program. This only applies to the character cell presentation.

Yes (Close CRT) – The UserExit Validation action is in use and the CRT (or window) is in its standard or *save* mode in the main program before the subroutine is called.

UserExit I/O

This field is similar to the above, except that it controls the UserExit Process action.

No – The UserExit Process action is not in use.

Yes (Don’t close CRT) – The UserExit Process action is in use and the CRT (or window) stays open in the main program. This only applies to the character cell presentation.

Yes (Close CRT) – The UserExit Process action is in use and the CRT (or window) is in its standard or *save* mode in the main program before the subroutine is called.

UserExit Screen Override

Indicate whether the system should check the Call UserExit field for each window to determine whether a call to the UserExit is to be performed for the window.

☞ *Note for VB or WEBdesk users:* This check box must be cleared so that the UserExit is implemented without regard to the value in the Call UserExit field for the individual window.

To eliminate the call to the UserExit from an individual window, set the Call UserExit field through Computron's Soft Technology facility. All windows default the Call UserExit field to blank or not applicable. Valid options are:

- Check the value in the Call UserExit field for each window to determine whether a UserExit call is to be performed.
- Implement the UserExit without regard to the value in the Call UserExit field for the individual window.

UserExit Revalidate Lines

This field determines whether all the lines of a ULM window should be revalidated via the UserExit just after selecting the Process/Save option. If used, the CRT (or window) status is controlled by the option entered for the UserExit Validations field, above. Valid options are:

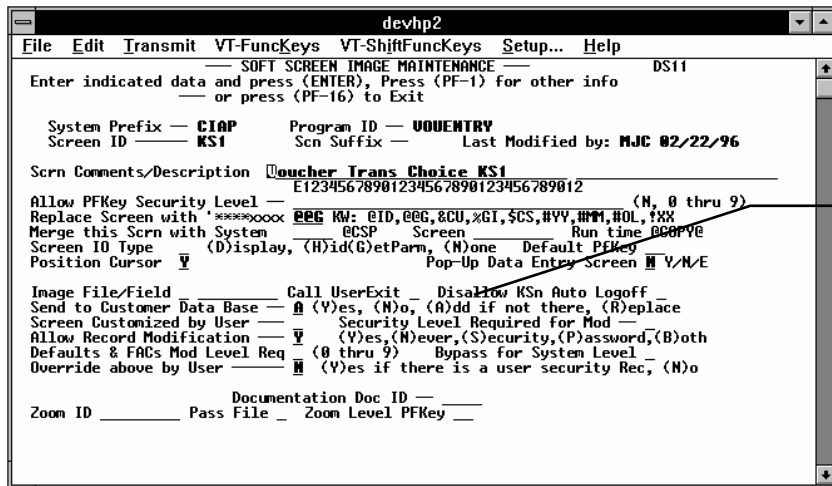
- Revalidate all the entered distribution lines after selecting the Process/Save option.
- Save the distribution lines without revalidating them.

☞ *Note:* This field is not available in all products' Global Parameter Maintenance functions.

7. Modify individual windows as appropriate. If the UserExit Screen Override field is selected in Global Parameters, you must modify the Call UserExit field setting for the individual windows where the UserExit is to be called.

Character Cell Presentation

Using the Soft Screens function, select the DR record of the window to be modified. Press the shortcut key F17, followed by Enter, to display the DS11 window, which is used to define overall window options.



Call UserExit field

Soft Screens Image Maintenance Window (DS11)

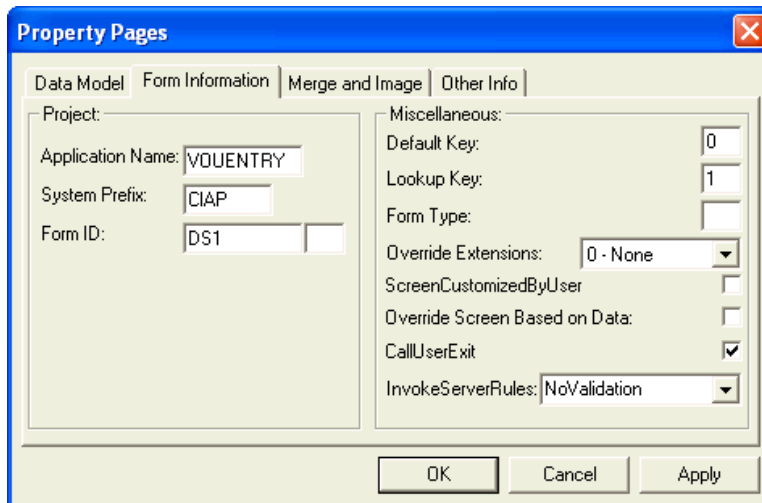
Call UserExit

This field must be set to Y (Yes) to enable the UserExit call from the selected window. A blank value in this field is the same as N (No) — no UserExit call is performed for the window.

Default: Blank or No.

VB or WEBdesk Presentations (using VB Forms Editor)

Use the Computron VB Forms Editor window editing utility to open the window that will call the UserExit. Then, click the Windows Parameters command on the Computron Options window.



Voucher Entry Window (DS1) – Windows Parameters

Call UserExit

This checkbox must be selected () to enable the UserExit call from the selected window. A blank checkbox means that no UserExit call is performed for the window.

Default:

Click Ok to save the change.

☞ *Note:* For detailed instructions on using Computron’s VB Forms Editor utility, refer to the *VB Forms Editor User’s Guide*.

Unix Implementation Using C

Introduction

The Computron run time utilizes Unix dynamically linked (shared) libraries to invoke user-definable object code. The directory `.../_ctron_/lib` (where `...` represents the file system where the Computron software resides) contains C source files, makefiles and the Computron supplied sample versions of the dynamically linked libraries. The dynamically linked libraries are usually called `libusersub.so`, `libusersub.sl` or `libusersub.o`, depending on the particular system.

Implementing a UserExit in C involves the following steps:

1. Copy the directory to a test-development location. Copy the directory `.../_ctron_/lib` (where `...` represents the directory name of the file system where the Computron software resides) to a test-development location.

☞ *Note:* Do not make any changes in `.../_ctron_/lib` while users are logged into the Computron application.

The following steps assume that the user is working in the test-development location.

2. Write the UserExit subroutine. The name for the subroutine must be `XXUEXIT`, where `XX` represents the product for which the UserExit is being written (`AP`, `AR`, `FA`, `GL`, `PO`, `TB`, etc.).
3. Enable run time to perform the custom UserExit subroutine. Add a code segment to the file `usersub.c` to enable the run time to intercept the call to the UserExit subroutine and perform the call to the UserExit subroutine created in step 2. Refer to the “Sample UserExit Templates” section at the end of this appendix for samples of the UserExit templates released with Computron software.
4. Modify the `makefile`. Add the custom UserExit to the dynamically linked library by modifying the `makefile` as follows:

```
USERSUBS=usersub.o
```

Add the name of the UserExit subroutine to the `USERSUBS` macro line.

Example:

Add `apuexit.o` to the line `USERSUBS=usersub.o` to create the line:

```
USERSUBS=usersub.o apuexit.o
```

5. Compile the UserExit subroutine. Compile the UserExit subroutine and build a new dynamically linked library containing the custom UserExit code via the `make` command (i.e., type `make`).

Refer to the description of the Unix `make` facility for more detailed information about the `make` command and its usage.

6. Alter the search path. Alter the search path for dynamically linked library files used by the Computron run time by changing a system environment parameter as described below. There are two possible ways to do this, a) or b).
 - a) Modify (or create) the file `/.ctron_start` in a user's home directory (`$HOME/.ctron_start`) to affect an individual user's search path. This does not apply to AUI users.
 - b) Modify (or create) the file `/.../_ctron_/local.start` (where `/...` represents the directory name of the file system where the Computron software resides) to affect the search path for all Computron users on the system.

The name of the directory where the dynamically linked library resides must be appended to the shared library search path

For AIX systems:

```
export LIBPATH=<location of library>:$LIBPATH
```

For HP-UX systems:

```
export SHLIB_PATH=< location of library >:$SHLIB_PATH
```

For all other systems:

```
export LD_LIBRARY_PATH=< location of library >:$LD_LIBRARY_PATH
```

7. Modify the Global Parameters. To modify the Accounts Payable parameters, run the Global Parameters Maintenance function (GLOBAPMT). Entries are required in the following fields:

UserExit Validations

This field determines whether the UserExit Validation action is performed, and, if so, whether the CRT (or window) remains open. Valid options are:

No – The UserExit Validation action is not in use.

Yes (Don't close CRT) – The UserExit Validation action is in use. In addition, the CRT (or window) stays open in the main program. This only applies to the character cell presentation.

Yes (Close CRT) – The UserExit Validation action is in use and the CRT (or window) is in its standard or *save* mode in the main program before the subroutine is called.

UserExit I/O

This field is similar to the above, except that it controls the UserExit Process action.

No – The UserExit Process action is not in use.

Yes (Don't close CRT) – The UserExit Process action is in use and the CRT (or window) stays open in the main program. This only applies to the character cell presentation.

Yes (Close CRT) – The UserExit Process action is in use and the CRT (or window) is in its standard or *save* mode in the main program before the subroutine is called.

UserExit Screen Override

Indicate whether the system should check the Call UserExit field for each window to determine whether a call to the UserExit is to be performed for the window.

☞ *Note for VB or WEBdesk users:* This check box must be cleared so that the UserExit is implemented without regard to the value in the Call UserExit field for the individual window.

To eliminate the call to the UserExit from an individual window, set the Call UserExit field through Computron's Soft Technology facility. All windows default the Call UserExit field to blank or not applicable. Valid options are:

- Check the value in the Call UserExit field for each window to determine whether a UserExit call is to be performed.
- Implement the UserExit without regard to the value in the Call UserExit field for the individual window.

UserExit Revalidate Lines

This field determines whether all the lines of a ULM window should be revalidated via the UserExit just after selecting the Process/Save option. If used, the CRT (or window) status is controlled by the option entered for the UserExit Validations field, above. Valid options are:

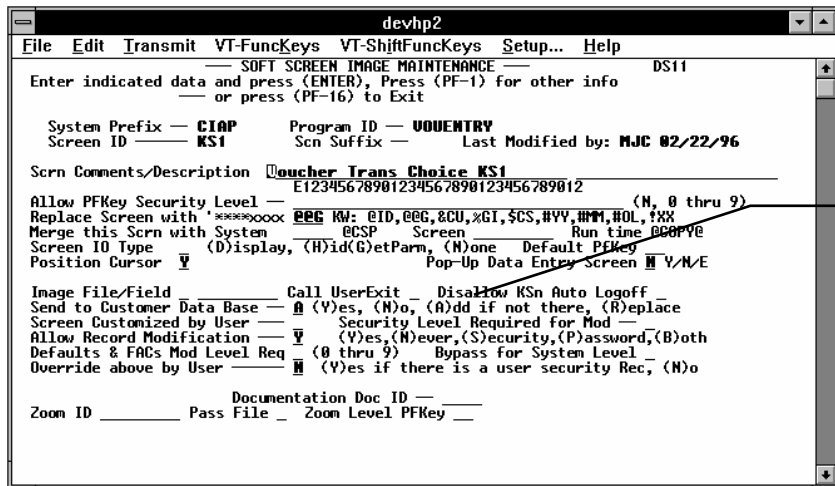
- Revalidate all the entered distribution lines after selecting the Process/Save option.
- Save the distribution lines without revalidating them.

☞ *Note:* This field is not available in all products' Global Parameter Maintenance functions.

8. Modify individual windows as appropriate. If the UserExit Screen Override field is selected in Global Parameters, you must modify the Call UserExit field setting for the individual windows where the UserExit is to be called.

Character Cell Presentation

Using the Soft Screens function, select the DR record of the window to be modified. Press the shortcut key F17, followed by Enter, to display the DS11 window, which is used to define overall window options.



Call UserExit field

Soft Screens Image Maintenance Window (DS11)

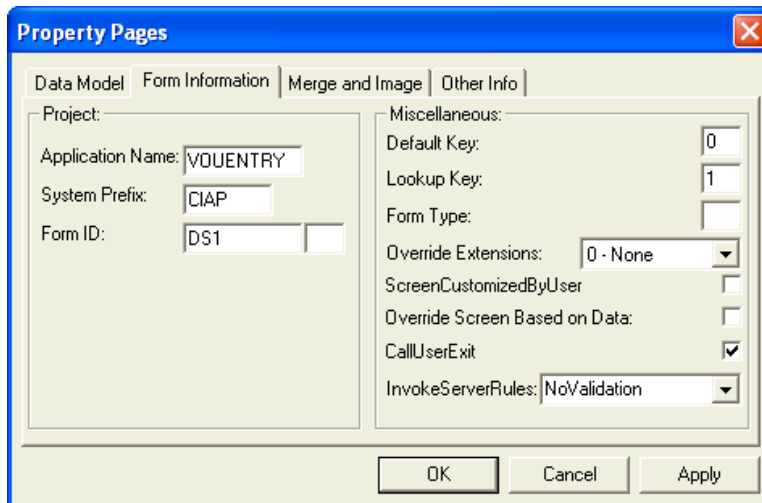
Call UserExit

This field must be set to Y (Yes) to enable the UserExit call from the selected window. A blank value in this field is the same as N (No) — no UserExit call is performed for the window.

Default: Blank or No.

VB or WEBdesk Presentations (using VB Forms Editor)

Use the Computron VB Forms Editor window editing utility to open the window that will call the UserExit. Then, click the Windows Parameters command on the Computron Options window.



Voucher Entry Window (DS1) – Windows Parameters

Call UserExit

This checkbox must be selected () to enable the UserExit call from the selected window. A blank checkbox means that no UserExit call is performed for the window.

Default:

Click Ok to save the change.

☞ *Note:* For detailed instructions on using Computron's VB Forms Editor utility, refer to the *VB Forms Editor User's Guide*.

Technical Description

Files Included

The following system components are provided upon request. They facilitate custom UserExit development.

XXUEXITD

Data Structures. This COBOL Copy Member describes the data structures needed for the UserExit. This copy member is available for each application system (replace XX with the appropriate system prefix: AP, AR, GL, etc.).

XXUEXITs

Shell COBOL UserExit subroutine. This subroutine provides the user with a basic UserExit subroutine, and includes documentation that helps MIS users to develop custom UserExits. The UserExit program should be renamed APUEXIT, ARUEXIT, GLUEXIT, FAUEXIT, TBUEXIT, etc. as appropriate.

../exec/_ctron/_lib

This Unix directory contains two files of object code which can be utilized to write UserExits under C environments.

Each of these components is described in greater detail on the following pages.

XXUEXITD

This standard copy member describes each of the data elements which are utilized during UserExit processing. These data elements define parameters used to control how the UserExit is processed. A sample of the APUEXITD file (XXUEXITD for the Accounts Payable system) is included on the following pages. Because this copy member can be modified by Computron at any time, it is recommended that the user review the most recent version of the file as it is included in the CVAPCOPY library on the local system. (Use Computron's Display utility (CSDISPLY) to view and print the APUEXITD file.)

The parameters which are passed to the UserExit include those described below. With the few exceptions indicated in these descriptions, the user does not set any of these values, but can read the values in order to perform conditional processing within the UserExit.

Screen Override Switch

This indicates whether the option to call the UserExit can be overwritten at the individual window level via Soft Technology. Valid values are:

Y – The option to call the UserExit can be overwritten at the individual window level.

N – UserExit calls cannot be overwritten at the individual window level.

Transfer Audit Switch

This parameter is reserved for future use.

Validate Switch

This field indicates whether UserExit processing is to be performed following execution of Computron validations. Valid values are:

N – None.

Y – Yes, CRT is open.

C – Yes, CRT is in standard mode.

Process Switch

This field indicates whether UserExit processing is to be performed when the user executes the option to update Computron files, i.e., Enter on the last window of a data entry program, the Save/Process option on a ULM window. Valid values are:

N – None.

Y – Yes, CRT is open.

C – Yes, CRT is standard mode.

CRT Active Switch

This field is set to Y if the CRT is open while the UserExit is in process.

CRT Closed Switch

This field is set to Y if the CRT is in standard mode while the UserExit is in process.

Screen Erased Switch

This switch is set by the UserExit and returned to the calling program if part or all of the original window is erased by the UserExit. This allows the calling program to refresh the entry window appropriately.

UserExit Action

This action acts as a trigger for the UserExit to complete its processing. Valid values are:

S – START

V – VALIDATE

B – BEFORE

P – PROCESS

T – TERMINATE

U – FINAL-VAL

Traffic Control

This data element contains the calling program name.

Call From Screen

This contains the ID of the window from which the UserExit was called or some unique ID within the program, if the call is not after a specific window.

File ID

This contains the name of the primary file being updated by the Computron program. In some Computron programs, multiple files can be updated. However, the UserExit is only passed the Record Hold area (see Record Hold below) of this file. The UserExit can open this or any other file to read or update (where allowed).

Record Key

This is the key to the current record being updated by the Computron program.

Mode

This data element contains the current processing mode. Valid values are:

- A – Add.
- C – Change.
- D – Delete

PF-Key

The option used to execute the most recent operation.

Return Message

Initially, this is set to blanks by the calling program. The UserExit can modify this value to return a message to the user. This message will be displayed in the message area of the Computron window upon return to the calling program. Processing cannot continue until this message is cleared. The message text can be up to 79 positions long.

Escape

When called with a Start action, this parameter can be set by the UserExit itself. If set to Y, the Computron program terminates immediately, without performing any updates. The value in this parameter is only checked upon return to the calling program following the Start action. This is useful in cases where the UserExit performs some type of security validation which extends beyond Computron's security subsystem.

Records Held Switch

This indicates whether the calling program is holding records. If this is set to Y, the UserExit must not update any records with hold, as it would confuse Computron's locking strategy and jeopardize data integrity.

- ☞ *Note:* Regardless of whether this switch is set to Y (Yes), the called UserExit should **never** perform updates except during an appropriate call to the Process action (The Process action is called by all programs after all Computron updates have been completed).

Company Number

This parameter is not currently used.

User Interface Area

This data area is included in every Computron file. This allows the user to define installation-specific data to maintain in the Computron system. Very frequently, UserExits are implemented to validate this data against external files. While no more than 255 positions are available for any file in Computron's systems, the UIA varies for each file. To determine the

amount of space available for the target file, the user can review the Data Description List for the file via the DDL Maintenance function.

Record Hold

This area includes all the data in the primary data file (see the File ID parameter above) accessed by the calling program. The called UserExit can access this data for validation. The user cannot modify any of this area.

The fields included vary from file to file. However, the redefinition of this area is interpreted from the corresponding Data Description List (in the CIXXCTL library, where XX represents the product). A maximum of 2024 positions are available for any Computron file. To determine the record size, review the Data Description List for the file via the DDL Maintenance function.

In-Hold

This area includes all the data in the program input area for the calling program. The called UserExit can access this data for validation. The user cannot modify any of the data in this area.

The fields included vary, depending upon the calling program. However, the redefinition of this area is interpreted from the corresponding Data Description List (in the CIXXCTLI library, where XX represents the application). A maximum of 3000 positions are available for any Computron file, with positions 2001 through 3000 reserved for use by ULM line items. To determine the record size, review the Data Description List for the file via the DDL Maintenance function.

```

000100*****ET270825
000200*   APUEXITD: *GL190524
000300*   A/P   USER EXIT DATA STRUCTURES *GL190524
000400*   COMPUTRON SYSTEMS CO., INC COBOL STANDARDS *ET270825
000500* *ET270825
000600* VERSION   DATE   MOD COMMENT *ET270825
000700* ----- *ET270825
000800* 02.01.00 08/25/87 ETT INITIAL RELEASE. *ET270825
000900* 02.02.00 05/23/89 ETT MISC FINAL MODS. *ET290523
001000* 02.02.01 11/19/90 GJL ADD FINAL ULM VALIDATE FUNCTION GL301119
001010* 02.02.02 05/07/91 ETT ADD EXTENDED DATA SW FOR INTERNAL USE ONLY.ETT10507
001020* 02.02.03 04/05/93 GJL NEW MAX NUM ITEMS CM 30405
001100*****ET270825
001200 ET270825
001300 01 APUEXIT-CONTROL-AREA. GL190524
001400 03 APUEXIT-CALL-UEXIT-SW. GL190524
001500 88 APUEXIT-CALL-UEXIT-NONE VALUE "NN", " ". GL190524
001600 05 APUEXIT-CALL-UEXIT-VALIDATE-SW PIC X VALUE "N". GL190524
001700* %VALSW GL210222
001800 88 APUEXIT-CALL-UEXIT-VALIDATE VALUE "Y", "C". GL190524
001900 88 APUEXIT-CALL-UEXIT-VALIDATE-C VALUE "C". GL190524
002000 05 APUEXIT-CALL-UEXIT-PROCESS-SW PIC X VALUE "N". GL190524
002100* %PROCSW GL210222
002200 88 APUEXIT-CALL-UEXIT-PROCESS VALUE "Y", "C". GL190524
002300 88 APUEXIT-CALL-UEXIT-PROCESS-C VALUE "C". GL190524
002400 03 APUEXIT-CALL-SCN-OVERRIDE-SW PIC X VALUE "N". GL190524
002500* %SCNOVRD GL210222
002600 88 APUEXIT-CALL-SCN-OVERRIDE VALUE "Y". GL190524
002700 03 APUEXIT-CALL-TRANSFER-AUDIT-SW PIC X VALUE "N". GL190524
002800* %TRANAUDT GL210222
002900 88 APUEXIT-CALL-TRANSFER-AUDIT VALUE "Y". GL190524
003000 03 APUEXIT-CALL-TO-VALIDATE-SW PIC X. GL190524
003100* %VDATE GL210222
003200 88 APUEXIT-CALL-TO-VALIDATE VALUE "Y", "C". GL190524
003300 88 APUEXIT-CALL-TO-VALIDATE-C VALUE "C". GL190524
003400 03 APUEXIT-CRT-ACTIVE-SW PIC X VALUE "Y". GL190524
003500* %CRTACT GL210222
003600* 88 APUEXIT-CRT-ACTIVE VALUE "Y". GL190524
003700 03 APUEXIT-CRT-IS-CLOSED-SW PIC X VALUE "N". GL190524
003800* %CRTCLS GL210222

```

003900		88	APUEXIT-CRT-IS-CLOSED		VALUE "Y".	GL190524
004000	03		APUEXIT-SCREEN-ERASED-SW		PIC X VALUE "N".	GL190524
004100*					%SCNERSE	GL210222
004200*		88	APUEXIT-SCREEN-ERASED		VALUE "Y".	GL190524
004300	03		APUEXIT-ACTION		PIC X(01) VALUE SPACES.	GL190524
004400*					%ACTION	GL210222
004500		88	APUEXIT-ACTION-START		VALUE "S".	GL190524
004600		88	APUEXIT-ACTION-VALIDATE		VALUE "V".	GL190524
004700		88	APUEXIT-ACTION-BEFORE		VALUE "B".	GL190524
004800		88	APUEXIT-ACTION-INFO		VALUE "I".	GL190524
004900		88	APUEXIT-ACTION-PROCESS		VALUE "P".	GL190524
005000		88	APUEXIT-ACTION-TERMINATE		VALUE "T".	GL190524
005100*B-02.02.01						*GL301119
005200		88	APUEXIT-ACTION-VAL-FINAL-ULM		VALUE "U".	GL301119
005300*E-02.02.01						*GL301119
005400	03		APUEXIT-TRAFFIC-CONTROL		PIC X(08) VALUE SPACE.	GL190524
005500*					%TRCNTRL	GL210222
005600	03		APUEXIT-CALL-FROM-SCREEN		PIC X(08) VALUE SPACE.	GL190524
005700*					%CALLFROM	GL210222
005800	03		APUEXIT-ID.			GL190524
005900		05	APUEXIT-FILE-ID		PIC X(08) VALUE SPACE.	GL190524
006000*					%FILEID	GL210222
006100		05	APUEXIT-REC-KEY		PIC X(50) VALUE SPACE.	GL190524
006200*					%RECKEY	GL210222
006300***			ABOVE IS NOT USED			ET290523
006400	03		APUEXIT-MODE		PIC X VALUE SPACE.	GL190524
006500*					%MODE	GL210222
006600*		88	APUEXIT-MODE-ADD		VALUE "A".	GL190524
006700*		88	APUEXIT-MODE-INSERT		VALUE "A", "I".	GL190524
006800*		88	APUEXIT-MODE-CHANGE		VALUE "C".	GL190524
006900*		88	APUEXIT-MODE-DELETE		VALUE "D".	GL190524
007000*		88	APUEXIT-MODE-REVIEW		VALUE "R".	GL190524
007100	03		APUEXIT-PF-KEY		PIC 9(02) VALUE ZEROES.	GL190524
007200*					%PFKEY	GL210222
007300	03		APUEXIT-MSG-SET.			GL210222
			*%MSGSET			
007400		05	APUEXIT-MSG-KEY-SYSTEM		PIC X(04) VALUE SPACES.	GL190524
007500						GL210222
			*%MSGSYS			
007600		05	APUEXIT-MSG-PROGRAM		PIC X(08) VALUE SPACES.	GL190524
007700						GL210222
			*%MSGPROG			
007800		05	APUEXIT-MSG-CODE		PIC X(08) VALUE SPACES.	GL190524
007900						GL210222
			*%MSGCODE			
008000		05	FILLER		PIC X(03) VALUE SPACES.	ET290523
008100		05	APUEXIT-MSG-DISP.			GL190524
008200		09	APUEXIT-MSG-FAC		PIC X(01) VALUE SPACES.	GL190524
008300		09	APUEXIT-MSG		PIC X(79) VALUE SPACES.	GL190524
008400						GL210222
			*%MESSAGE			
008500		05	APUEXIT-MSG-FIELD-OCCURENCE		PIC 99 VALUE ZERO.	GL190524
008600						GL210222
			*%FLDOCCUR			
008700		05	APUEXIT-MSG-FIELD-WHAT-FILE		PIC X VALUE SPACES.	GL190524
008710						GL210222
			*%FLDFILE			
008800		05	APUEXIT-MSG-DRIVER-READ-IT		PIC X VALUE " "	GL190524
008801**			THE FOLLOWING IS USED IN GENEX PROGRAMS SO THAT THE VALUE			GL 30806
008802**			THAT CAUSED THE ERROR CAN BE PASSED BACK & LATER REPORTED			GL 30806
008810		03	APUEXIT-ERR-VALUE REDEFINES APUEXIT-MSG-SET			GL 30806
008820					PIC X(12).	GL 30806
008900		03	APUEXIT-ESCAPE-SW		PIC X VALUE "N".	GL190524
008910						GL210222
			*%ESCAPESW			
009000		88	APUEXIT-ESCAPE		VALUE "Y".	GL190524
009100		03	APUEXIT-RECORDS-ARE-HELD-SW		PIC X VALUE "N".	GL190524
009110						GL210222
			*%RECSHLD			
009200***			ABOVE SHOULD BE EXTRACTED IN USER EXIT			ETT80114
009300*		88	APUEXIT-RECORDS-ARE-HELD		VALUE "Y".	GL190524
009400*						ET270825
009500		03	APUEXIT-COMP-NUM		PIC X(03) VALUE SPACES.	GL110619
009600***			ABOVE NOT USED			ET290523
009601*B-02.02.02						*ETT10507

```

009610      03  APUEXIT-EXTENDED-DATA-SW      PIC X          VALUE "N".      ETT10507
009620***  INTERNAL USE ONLY!!! DO NOT MODIFY VALUE OF FIELD ABOVE      ETT10507
009630      03  APUEXIT-ULM-CUR-ITEM          PIC S9(5) COMP-3      GL130201
009630                                          VALUE 0.              GL130201
009640***  IF CALLED FOR VALIDATE FROM ULM, WILL CONTAIN THE OCCURENCE      GL130201
009650***  NUMBER OF THE ITEM BEING VALIDATED.                            GL130201
009655*B-0?..??..??                          *GL 30224
009660      03  APUEXIT-TB-INV-DATE-FLAG      PIC X(01)      VALUE "N".      GL 30224
009661***  IF SET TO 'Y', A CHANGE OF INVOICE DATE IN VOUCHANG WILL CAUSEGL 30224
009662***  APTBUEXT TO SEND REVERSALS AND ADDS FOR ALL THE LINES.        GL 30224
009665*E-0?..??..??                          *GL 30224
009670*B-02.02.03                             *CM 30405
009675      03  APUEXIT-UL1-MAX-ITEMS        PIC S9(5) COMP-3      CM 30405
009675                                          VALUE 0.              CM 30405
009700      03  FILLER                        PIC X(092) VALUE SPACES.  CM 30405
009701*E-02.02.03                             *CM 30405
009710*E-02.02.02                             *ETT10507
009800      03  APUEXIT-USER-INTERFACE-AREA  PIC X(255) VALUE SPACES. GL190524
009900***  BEFORE CALLING UEXIT, MOVE USER INTERFACE AREA TO ABOVE      ET 80603
010000***  AND WHEN YOU COME BACK, MOVE THE ABOVE AREA TO USER INTERFACE ET 80603
010100      03  APUEXIT-USER-INTERFACE-AREA60 REDEFINES                    GL190524
010200          APUEXIT-USER-INTERFACE-AREA PIC X(060).                  GL190524
010210                                          GL210222
          *%USER60
010300      03  APUEXIT-USER-INTERFACE-AREA120 REDEFINES                    GL190524
010400          APUEXIT-USER-INTERFACE-AREA PIC X(120).                  GL190524
010410                                          GL210222
          *%USER120
010500      03  APUEXIT-USER-IFACE-REGISTER  REDEFINES                    GL190930
010600          APUEXIT-USER-INTERFACE-AREA.                            GL190930
010700      05  APUEXIT-USER-IFACE-REGISTER-1 PIC X(30).                  GL390930
010710                                          GL210222
          *%USER30
010800      05  APUEXIT-USER-IFACE-REGISTER-2 PIC X(160).                GL190930
010810                                          GL210222
          *%USER160
010900*                                          ET270825
011000 01  APUEXIT-IN-DATA.                                                GL190524
011100      03  APUEXIT-RECORD-HOLD          PIC X(2048) VALUE SPACES.  GL190524
011200      03  APUEXIT-IN-HOLD             PIC X(3000) VALUE SPACES.  GL190524
011300*                                          ET270825

```

XXUEXITS

The XXUEXITS shell UserExit is provided upon request. The following pages provide a listing of this file for reference purposes. Notice that instructions are included as comment lines (a * in position 5) in the file. When users create custom UserExits, this file is copied to a new name (usually XXUEXIT where XX is the appropriate system prefix) and modified to include the code which will perform the desired UserExit processing. This file is a generic file for all applications. Changes may have to be made where a line of code identifies a literal in quotations (e.g., line 013800 reads IF ??UEXIT-TRAFFIC-CONTROL = "ACCTMNT") or a particular program (e.g., line 014100 reads PERFORM TENTRY) to reflect the UserExit being developed.

The sections listed below are to be modified in the user's copy of XXUEXITS with the appropriate information.

```

000100***      1. REPLACE ??UEXIT WITH THE USER EXIT NAME                  ET280225
000200***          SUCH AS ARUEXIT, GLUEXIT, APUEXIT ETC                  ET280225
000300***      2. REPLACE ?SYS WITH THE SYSTEM PREFIX SUCH AS              ET280225
000400***          CIGL, CIAR, CIAP ETC                                  ET280225
000500***      3. UPDATE ALL COMMENTS AND CLIENT INFO (???????????)      ET280225
000600/*****ET280225
000700*                                          *ET280225
000800*          ??UEXIT                                                  *ET280225
000900*                                          *ET280225
001000*****ET280225
001100*                                          *ET280225
001200*      COMPUTRON SOFTWARE                                          *ET280225

```

```

001300*          A DIVISION OF COMPUTRON TECHNOLOGIES CORPORATION          *ET280225
001400*          301 ROUTE 17 NORTH                                         *ET 10107
001500*          RUTHERFORD, NEW JERSEY  07070                             *ET 10107
001600*          TEL: (201) 935-3400                                       *ET 10107
001700*          FAX: (201) 935-6355                                       *ET 10107
001800*          *****                                                    *ET280225
001900*          *****                                                    *ET280225
002000*          PROGRAM:  ??UEXIT                                           *ET280225
002100*          SYSTEM:   ?SYS                                             *ET280225
002200*          CLIENT:   ????????????                                     *ET280225
002300*          FUNCTION:  USER-EXIT SUBROUTINE.                          *ET280225
002400*          *****                                                    *ET280225
002500*          RELEASE HISTORY:                                           *ET280225
002600*          *****                                                    *ET280225
002700*          LEVEL      DATE      REP      DESCRIPTION                    *ET280225
002800*          -----      -
002900*          02.01.00  02/25/88  ETT      INITIAL RELEASE.                  *ET280225
003000*          *****                                                    *ET280225
003100*          *****                                                    *ET280225
003200*          *****                                                    *ET280225
003300*          *****                                                    *ET280225
003400*          *****                                                    *ET280225
003500*          *****                                                    *ET280225
003600*          *****                                                    *ET280225
003700 IDENTIFICATION DIVISION.                                           ET280225
003800 PROGRAM-ID.                                                         ET280225
003900    ??UEXIT.                                                         ET280225
004000 AUTHOR.                                                             ET280225
004100    ELIAS TYPALDOS.                                                  ET280225
004200 DATE-WRITTEN.                                                       ET280225
004300    02/25/88.                                                       ET280225
004400 DATE-COMPILED.                                                     ET280225
004500/*****                                                                *ET280225
004600*          *****                                                    *ET280225
004700*          ENVIRONMENT DIVISION                                       *ET280225
004800*          *****                                                    *ET280225
004900/*****                                                                *ET280225
005000*          *****                                                    *ET280225
005100 ENVIRONMENT DIVISION.                                               ET280225
005200 CONFIGURATION SECTION.                                             ET280225
005300 SOURCE-COMPUTER.                                                  ET280225
005400    RS6000.                                                         ET280225
005500 OBJECT-COMPUTER.                                                  ET280225
005600    RS6000.                                                         ET280225
005700 SPECIAL-NAMES.                                                    ET280225
005800/*****                                                                *ET280225
005900*          *****                                                    *ET280225
006000*          INPUT-OUTPUT SECTION                                       *ET280225
006100*          *****                                                    *ET280225
006200/*****                                                                *ET280225
006300*          *****                                                    *ET280225
006400*INPUT-OUTPUT SECTION.                                             ET280225
006500*FILE-CONTROL.                                                      ET280225
006600          *****                                                    *ET280225
006700          *****                                                    *ET280225
006800/*****                                                                *ET280225
006900*          *****                                                    *ET280225
007000*          DATA DIVISION - FILE SECTION                              *ET280225
007100*          *****                                                    *ET280225
007200/*****                                                                *ET280225
007300*          *****                                                    *ET280225
007400 DATA DIVISION.                                                    ET280225
007500*FILE SECTION.                                                      ET280225
007600*          *****                                                    *ET280225
007700*          *****                                                    *ET280225
007800/*****                                                                *ET280225
007900*          *****                                                    *ET280225
008000*          WORKING STORAGE                                           *ET280225
008100*          *****                                                    *ET280225
008200/*****                                                                *ET280225
008300*          *****                                                    *ET280225
008400 WORKING-STORAGE SECTION.                                           ET280225
008500 01  VERSION-DATA          PIC X(08) VALUE "02.01.00".             ET280225
008600          *****                                                    *ET280225
008700 01  SWITCHES.                                                     ET280225

```



```

008800      03  FIRST-TIME-SW          PIC X VALUE "Y".          ET280225
008900      88  FIRST-TIME              VALUE "Y".          ET280225
009000      03  FILES-ARE-OPEN-SW     PIC X VALUE "N".          ET280225
009100      88  FILES-ARE-OPEN       VALUE "Y".          ET280225
009200*                                           ET280225
009300*****ET280225
009400*      FOR COMPUTRON SPECIFIC ROUTINES ONLY          *ET280225
009500*****ET280225
008700*      COPY "UNXSTD"      IN CSUVCOPY.          ET280225
009600*      COPY GTUSERD2 IN CSUVCOPY.          ET280225
009700*      COPY OPENFLD2 IN CSUVCOPY.          ET280225
009800*      COPY STDITEMS IN CSUVCOPY.          ET280225
009900*****ET280225
010000*                                           ET280225
010100*                                           ET280225
010200 LINKAGE SECTION.          ET280225
010300                                           ET280225
010400*****ET280225
010500      COPY "??UEXITD" IN ?SYSCOPY.          ET280225
010600*****ET280225
010805 01  SCN-CONTROL-AREA          PIC X.          ETT10221
010810 01  SCN-ACTION-CODE          PIC X.          ETT10221
010815 01  CRT                      PIC X.          ETT10221
010820 01  CRTREC                  PIC X.          ETT10221
010825 01  SCNIMAGE-FILE-CONTROL    PIC X.          ETT10221
010830 01  SCN-IN-DATA              PIC X.          ETT10221
010835 01  SCN-DISPLAY-HOLD1        PIC X.          ETT10221
010840 01  SCN-DISPLAY-HOLD2        PIC X.          ETT10221
010845 01  SCN-DISPLAY-HOLD3        PIC X.          ETT10221
010850 01  SCN-DISPLAY-HOLD4        PIC X.          ETT10221
010855 01  SCN-DISPLAY-HOLD5        PIC X.          ETT10221
010860 01  SCN-GLOBAL-VALUES-AREA   PIC X.          ETT10221
010865 01  SCN-PROG-TITLES          PIC X.          ETT10221
010870 01  SCN-RETURN-MSG-SET       PIC X.          ETT10221
010875 01  SCX-OPERATOR-LEVEL       PIC X.          SBD10329
010880 01  SCN-NUM-SCREENS          PIC X.          ETT10221
010885 01  SCX-CURRENT-PROGRAM-NAME-INFO PIC X.          SBD10329
010890 01  SCN-LAST-ACTION-CODE     PIC X.          ETT10221
010900*****ET280225
011000                                           ET280225

```

Depending upon the volume of code required to perform the desired UserExit action, the Procedure Division can be constructed in one of the following two ways:

Alternative #1:

All UserExit actions for all Computron programs, that are to trigger UserExit actions, are included in XXUEXIT. The MAIN LOGIC section of the Procedure Division contains IF ELSE statements to determine the calling program. This calling program redirects processing to a program specific section identified by the program name. This is the manner in which the shell UserExit is written.

Alternative #2:

The MAIN LOGIC section of the Procedure Division determines the calling program and calls another subroutine, external to XXUEXIT. Note that external subroutines need to be linked or included in sharable images or dynamically linked libraries, etc.

```

011000                                           ET280225
011100/*****ET280225
011200*                                           *ET280225
011300*      PROCEDURE DIVISION          *ET280225
011400*                                           *ET280225
011500*****ET280225
011600 PROCEDURE DIVISION USING  ??UEXIT-CONTROL-AREA,          ET280225
011700                                ??UEXIT-IN-DATA,          ET280225
011701*** PARAMETERS BELOW ARE FOR COMPUTRON PROGRAMS ONLY          ETT10221
011702*** AND MAY NOT BE MODIFIED BY USER EXIT !!!!!          ETT10221
011710                                SCN-CONTROL-AREA,          ETT10221
011720                                SCN-ACTION-CODE,          ETT10221
011800                                CRT,          ET280225

```

```

011900 CRTREC, ETT10221
011901 SCNIMAGE-FILE-CONTROL, ETT10221
011910 SCN-IN-DATA, ETT10221
011920 SCN-DISPLAY-HOLD1, ETT10221
011930 SCN-DISPLAY-HOLD2, ETT10221
011940 SCN-DISPLAY-HOLD3, ETT10221
011950 SCN-DISPLAY-HOLD4, ETT10221
011955 SCN-DISPLAY-HOLD5, ETT10221
011960 SCN-GLOBAL-VALUES-AREA, ETT10221
011961 SCN-PROG-TITLES, ETT10221
011965 SCN-RETURN-MSG-SET, ETT10221
011970 SCX-OPERATOR-LEVEL, SBD10329
011975 SCN-NUM-SCREENS, ETT10221
011980 SCX-CURRENT-PROGRAM-NAME-INFO, SBD10329
011982 SCN-LAST-ACTION-CODE. ETT10221
012000*****ET280225
012100* WARNING!!!: THIS ROUTINE MUST NOT MODIFY ANY VALUES ET280225
012200* IN ??UEXIT-IN-DATA AREA ET280225
012300* OTHERWISE UNPREDICTABLE RESULTS MAY OCCUR ET280225
012400* WITHIN THE MAIN COMPUTRON PROGRAM. ET280225
012500* WARNING!!!: NO FILES UPDATES ARE ALLOWED UNLESS ET290821
012600* ??UEXIT-ACTION-START, ET290821
012610* ??UEXIT-ACTION-PROCESS, ET290821
012620* ??UEXIT-ACTION-TERMINATE. ET290821
012700* ANY FILE UPDATES MAY CAUSE FILE MANAGEMENT ET 10107
012710* TO RESPOND WITH UNPREDICTABLE RESULTS IN ET 10107
012900* THE COMPUTRON PROGRAM UNDER CERTAIN CIRCUMSTANCES ET 10107
013000*****ET280225
013100* ET280225
013200 MAIN-LOGIC SECTION. ET280225
013300 MAIN-LOGIC-S. ET280225
013500 IF FIRST-TIME ET280225
013600 PERFORM PROGRAM-INITIALIZATIONS. ET280225
013700 ET280225
013800 IF ??UEXIT-TRAFFIC-CONTROL = "ACCTMNT" ET280225
013900 PERFORM ACCTMNT ET280225
014000 ELSE IF ??UEXIT-TRAFFIC-CONTROL = "TENTRY" ET280225
014100 PERFORM TENTRY ET280225
014200 ELSE ET280225
014300 NEXT SENTENCE. ET280225
014400 ET280225
014500 MAIN-LOGIC-END. ETT80225
014600 EXIT PROGRAM. ET280225
014700 MAIN-LOGIC-EXIT. ET280225
014800 EXIT. ET280225
014900*****ET280225
015000 PROGRAM-INITIALIZATIONS SECTION. ET280225
015100 PROGRAM-INITIALIZATIONS-S. ET280225
015200* FOR COMPUTRON ROUTINES ONLY ET280225
015300* PERFORM GET-USER. ET280225
015400*** MAY DO EXTRACTS, OR OPEN COMMON FILES ETC. ET280225
015500 MOVE "N" TO FIRST-TIME-SW. ET280225
015600/*****ET280225

```

For each Computron program that calls the UserExit, the lines of code performed are entered in the following form. First establish the type of action (Start, Validate, Process, Terminate, etc.). Then, for each type of action call, enter the lines of code that identify the desired user-defined process applied at that point in the program.

```

015600/*****ET280225
015700 ACCTMNT SECTION. ET280225
015800 ACCTMNT-S. ET280225
015900 IF ??UEXIT-ACTION-START ET280225
016000 PERFORM ACCTMNT-START ET280225
016100 ELSE IF ??UEXIT-ACTION-VALIDATE ET280225
016200 PERFORM ACCTMNT-VALIDATE ET280225
016300 ELSE IF ??UEXIT-ACTION-INFO ET280225
016400 PERFORM ACCTMNT-INFO ET280225
016500 ELSE IF ??UEXIT-ACTION-PROCESS ET280225
016600 PERFORM ACCTMNT-PROCESS ET280225
016700 ELSE IF ??UEXIT-ACTION-TERMINATE ET280225

```

```

016800          PERFORM ACCTMNT-TERMINATE          ET280225
016900          ELSE                              ET280225
017000          GO TO ACCTMNT-EXIT.                ET280225
017100 ACCTMNT-EXIT.                              ET280225
017200          EXIT.                              ET280225
017300*****                                     ET280225
017400 ACCTMNT-START SECTION.                     ET280225
017500 ACCTMNT-START-S.                           ET280225
017600*** COMES HERE RIGHT AFTER THE COMPUTRON PROGRAM HAS OPENED ET280225
017700*** PROGRAM FILES AND PROGRAM INITIALIZATION HAS BEEN PERFORMED ET280225
017800*** THE ONLY FIELD SET AT THIS POINT IS ??UEXIT-TRAFFIC-CONTROL. ET280225
017900*** YOU MAY SET THE ??UEXIT-ESCAPE-SW TO 'Y' IN ORDER TO FORCE ET280225
018000*** THE COMPUTRON PROGRAM TO TERMINATE JUST BY CLOSING THE FILES ET280225
018100*** IF THE ESCAPE-SW IS 'Y' THIS ROUTINE WILL NOT BE CALLED AGAIN. ET280225
018200*** YOU MAY ALSO SET ??UEXIT-CALL-UEXIT-VALIDATE-SW TO 'N' ET280225
018300*** IF YOU DON'T NEED THE ROUTINE TO BE CALLED AFTER EACH ET280225
018400*** SCREEN VALIDATION                       ET280225
018500*** OR YOU MAY ALSO SET ??UEXIT-CALL-UEXIT-PROCESS-SW TO 'N' ET280225
018600*** IF YOU DON'T NEED THE ROUTINE TO BE CALLED AFTER EACH ET280225
018700*** MAIN COMPUTRON PROGRAM TRANSACTION UPDATE ET280225
018800          MOVE "Y" TO FILES-ARE-OPEN-SW.     ET280225
018900*****                                     ET280225
019000 ACCTMNT-VALIDATE SECTION.                   ET280225
019100 ACCTMNT-VALIDATE-S.                         ET280225
019200*** COMES HERE AFTER EACH SCREEN HAS BEEN DISPLAYED AND ACCEPTED ET280225
019300*** AND ALL COMPUTRON VALIDATIONS HAVE PASSED WITH NO ERROR ET280225
019400*** ANY VALUE IN ??UEXIT-RETURN-MSG WILL FORCE THE SCREEN ET280225
019500*** TO BE REDISPLAYED AND THE ERROR TO APPEAR ON THAT SCREEN ET280225
019600*** THE ABOVE ERROR MUST HAVE A VALUE OTHER THAN SPACES ET280225
019700*** IN AT LEAST ONE OF THE FIRST TWO CHARACTERS ET280225
019800*** AT THIS POINT THE DATA IS IN THE ??UEXIT-IN-DATA ET280225
019900*** THE LAYOUT OF WHICH IS DESCRIBED BY THE CONTROL FILE ET280225
020000*** HAVING THE SAME NAME AS THE PROGRAM IN ?SYSCTLI ET280225
020100*** AND THE CONTROL FILE HAVING THE SAME NAME AS THE FILE ET280225
020200*** IN ??UEXIT-FILE-ID IN ?SYSCTL. ET280225
020300*** ??UEXIT-FILE-ID MAY BE EXAMINED FURTHER FOR PROGRAMS ET280225
020400*** THAT PROCESS MULTIPLE FILES SO THAT YOU CAN PERFORM ET280225
020500*** THE APPROPRIATE TESTS. ET280225
020600*** ALSO ??UEXIT-REC-KEY HAS THE VALUE OF THE PRIMARY KEY, ET280225
020700*** ??UEXIT-MODE HAS THE MODE (A/I/C/D/R) ET280225
020800*** ??UEXIT-PF-KEY HAS THE PFKEY THAT WAS PRESSED ET280225
020900*** BY THE USER ON THE CURRENT SCREEN. ET280225
021000*****                                     ET280225
021100*** NO FILE UPDATES ARE ALLOWED TO TAKE PLACE HERE ET280225
021200*** SINCE ANY FILE UPDATED WILL INVALIDATE THE HELD RECORDS ET280225
021300*** IN THE COMPUTRON PROGRAM AND THE RESULTS FROM DMS ET280225
021400*** WILL BE UNPREDICTABLE !!!!!!! ET280225
021500*****                                     ET280225
021600          IF ??UEXIT-CALL-FROM-SCREEN = "KS1" ET280225
021700          NEXT SENTENCE                       ET280225
021800          ELSE IF ??UEXIT-CALL-FROM-SCREEN = "DS1" ET280225
021900          NEXT SENTENCE                       ET280225
022000          ELSE                                ET280225
022100          GO TO ACCTMNT-VALIDATE-EXIT.         ET280225
022200*                                             ET280225
022300*          IF ??UEXIT-FILE-ID = "DIMREF" ET280225
022400*          MOVE ??UEXIT-REC-KEY TO ... ET280225
022500*          ELSE IF ??UEXIT-FILE-ID = "DIMATTR" ET280225
022600*          MOVE ??UEXIT-REC-KEY TO ... ET280225
022700*          ELSE                                ET280225
022800*          GO TO ACCTMNT-VALIDATE-EXIT.         ET280225
022900*                                             ET280225
023000*          IF ??UEXIT-MODE-CHANGE OR ??UEXIT-MODE-DELETE ET280225
023100*          MOVE "ERROR ... " TO GLUEXIT-RETURN-MSG ET280225
023200*          GO TO ACCTMNT-VALIDATE-EXIT.         ET280225
023300*                                             ET280225
023400 ACCTMNT-VALIDATE-EXIT.                       ET280225
023500          EXIT.                              ET280225
023600*****                                     ET280225
023700 ACCTMNT-INFO SECTION.                         ET280225
023800 ACCTMNT-INFO-S.                              ET280225
024200*** ALL OPTIONS AS IN VALIDATE ABOVE ARE PRESENT ET280225
024300*** EXCEPT ??UEXIT-PF-KEY IS NOT VALID ET280225
024400 ACCTMNT-INFO-EXIT.                          ET280225
024500          EXIT.                              ET280225

```

```
024600***** ET280225
024700 ACCTMNT-PROCESS SECTION. ET280225
024800 ACCTMNT-PROCESS-S. ET280225
024900*** COMES HERE ONLY AFTER ALL COMPUTRON PROGRAM UPDATES HAVE ET 10107
025000*** TAKEN PLACE FOR THE ENTIRE LOGICAL TRANSACTION, ET280225
025100*** AND A FREE ALL HAS BEEN ISSUED ET280225
025200*** AT THIS POINT YOU MAY CONSIDER ALL YOUR UPDATES ET280225
025300*** ANY VALUE IN ??UEXIT-RETURN-MSG WILL BE DISREGARDED ET280225
025400 ACCTMNT-PROCESS-EXIT. ET280225
025500 EXIT. ET280225
025600***** ET280225
025700 ACCTMNT-TERMINATE SECTION. ET280225
025800 ACCTMNT-TERMINATE-S. ET280225
025900*** COMES HERE AT THE END OF THE PROGRAM AND RIGHT BEFORE ET280225
026000*** THE COMPUTRON PROGRAM FILES ARE CLOSED. ET280225
026100*** ANY VALUE IN ??GLUEXIT-RETURN-MSG IS DISREGARDED ET 10107
026200/***** ET280225
026300* FOR COMPUTRON ROUTINES ONLY ET280225
026400* COPY OPENFLP2 IN CSUVCOPY. ET280225
026500* COPY GTUSERP2 IN CSUVCOPY. ET280225
*** End of Listing*****
```

Appendix C: LGMAP File

Introduction to LGMAP File

The LGMAP file maps volume names called by Computron systems to an actual physical location within the Unix file structure (i.e., paths). All volumes defined within a Computron system must be entered in the LGMAP file. Volume names are then used by the Database Map File Maintenance function.

LGMAP File Description

The LGMAP file is located in `.../exec/_ctron_/wisp/config` (where `...` represents the file system where the Computron software resides). The structure of the file is as follows:

- ◆ Each line defines one volume;
- ◆ The volume name can comprise up to six characters of the line;
- ◆ The seventh (7th) character of the line is always a space;
- ◆ The absolute path to the directory represented by this volume constitutes the remainder of the line;

Example:

```
Volume UNIX Path
PGMVOL /ctron/exec
IPLVOL /ctron/exec
EXEC   /ctron/exec
VOL777 /ctron/data
DATA   /ctron/data
```

- ◆ *Volume* can also be used to identify a Unix server or a database server. In this case, the server name is entered in lieu of the absolute path.

Example:

```
Volume UNIX Path
server ctron_database
```

Remote machines that act as servers are defined in the `.file_server` file, which is located in `.../exec/_ctron_` (where `...` represents the file system where the Computron software resides). A remote machine is denoted in the LGMAP file using the following format:

```
@ HostName:@Volume or @HostName:/pathname
```


Where `HostName` is the remote system and `Volume` is a volume name defined in the LGMAP file on the remote system. Alternatively, the Unix absolute pathname on the remote system can be specified.

Example:

```
Volume UNIX Path
server @Host0:VOL777
serv2  @Host1:/ctron/data
```

LGMAP Maintenance

A visual editor (like `vi`), can be used to change the contents of the LGMAP file. When adding new volumes or changing previously defined volumes, it is important to note that the path defined by the volume, library and filename called by a Computron application is restricted to eighty characters. For this reason, Computron recommends that the absolute path that defines a volume not exceed sixty-three characters.

 *Warning: Once the system is up and running, Computron strongly advises against changing existing LGMAP entries. If you need to add another volume name with the same or similar path to an existing entry, add a new one instead of changing an existing one.*

If the LGMAP file is changed, the program `wsysconf` must be run. This program creates a configuration file of the system containing information regarding the logical disk volumes and hardware setup on the system. Run the program from the Unix system prompt by entering the `wsysconf` command.

The system prompts for the default values as follows:

```
"RUNNING THIS PROGRAM WILL REPLACE THE PREVIOUS VERSION OF THE
FILE .../WISP/CONGIF/WSYSCONFIG"
DO YOU WISH TO CONTINUE (Y/N):
USE AUTOMATIC CONFIGURING (Y/N)?"
```

Respond `Y` to both questions.

Appendix D: WISP Command Processor

The WISP Command Processor is available for *only* character cell presentations on all Unix-based systems. The WISP Command Processor HELP menu is invoked by pressing `ctrl+e` from any point within Computron software. Briefly, this document explains how to manage accessibility to the WISP Command Processor functions that you want to make available to your users.

☞ *Note:* For Unix implementations using non-WISP terminals (e.g., terminals using foreign characters), pressing `ctrl+e` simply invokes a Unix shell.

Individual User Modifications

To restrict WISP Command Processor access for an individual user, you must first create the `.ctrtron_start` file in the user's HOME directory. Next, decide what level of restriction you want to place on the user and then add one of the following `wusage` commands.

- ◆ To deny access to the entire WISP Command Processor HELP menu, enter:

```
wusage flags set HELP=N
```

- ◆ To restrict access to the COMMANDS option (F11), enter:

```
wusage flags set COMMANDS=N
```

- ◆ To restrict access to the FILES/LIBRARIES option (F5), enter:

```
wusage flags set MODFILES=N
```

Global Modifications

If global changes are required for groups of individuals, then you need to modify the `local.start` file in the `/.../exec/_ctrtron_` directory (where `/...` represents the file system in which the Computron application software resides). The `local.start` file is automatically invoked by Computron's `start` script (which is executed by the user's `.profile` upon log in), and (if it exists) executes its contents.

🔔 *Important!* Computron does not advise modification of the `start` script because future releases of the software may inadvertently impact your changes.

1. Open the `local.start` file. If the file doesn't exist, create it using a Unix editor (e.g., `vi`).

☞ *Note:* Computron does not release the `local.start` file, since it is only used by clients who want to customize the `start` script.

2. Set the `local.start` file permissions to:

```
chmod = 774
owner = ctronsys
group = ctron
```

3. Decide to what degree you want to implement restricting the `wusage` functionality for a given user or set of users. You also need to add shell programming logic to the `local.start` file to complete this task.

Example 1:

To deny access to the entire WISP Command Processor HELP menu.

```
wusage flags set HELP=N
```

Example 2:

To remove access to the COMMANDS option (F11) for a specific user:

```
if [ "$LOGNAME" = john ]; then
    wusage flags set COMMANDS=N
fi
```

Example 3:

To remove access to the FILES/LIBRARIES option (F5) for a specific user:

```
if [ "$LOGNAME" = john ]; then
    wusage flags set MODFILES=N
fi
```

Example 4:

To create allow/disallow files that remove the HELP menu and/or the FILES/LIBRARIES / COMMANDS options for a group of users:

```
if fgrep -x "LOGNAME"/usr/local/lib/ctron_modno>/dev/null ;
then
    wusage flags set MODFILES=N
elif fgrep -x "LOGNAME"/usr/local/lib/ctron_helpno>/dev/null ;
then
    wusage flags set COMMANDS=N
elif fgrep -x "LOGNAME"/usr/local/lib/ctron_cmdno>/dev/null ;
then
    wusage flags set HELP=N
fi
```


WISP Command Constants

The following commands permit the setting and extract of usage constants.

wusage Command	Definition
wusage shell	Command Processor Shell.
wusage read [file]	Load usage constants from PERSONALITY.
wusage write [file]	Save usage constants to PERSONALITY.
wusage set <item>=<value>	Set a usage constant.
wusage extract <item>	Extract a usage constant.
wusage flags set <flag>=<logical>	Set one flag in the bit mask.
wusage version	Print the WISP version information.

<item>

INLIB, INVOL, OUTLIB, OUTVOL, RUNLIB, RUNVOL, SPOOLIB, SPOOLVOL, WORKLIB, WORKVOL, PROGLIB, PROGVOL, PRNTMODE, PRTCLASS, FORM#, PRINTER, JOBQUEUE, JOBCLASS, JOBLIMIT, FLAGS, USERID, LUSERID, LINES

<flag>

HELP, SETFILE, SETPRINT, SETSUB, PRTQUE, SYSTEM, UTILS, DISPLAY, EDIT, DISPRINT, CRID, TERMINAL, PSUEDO, CURSOR, SCREEN, COMMANDS, SAVE, PRINTSCR, CANCEL, FILES, MODFILES

<logical>

Y = yes

N = no

T = true

F = false

Examples

Set INLIB to equal MYLIB.

```
wusage set inlib=mylib
```

Extract RUNVOL into a shell variable.

```
shvar='wusage extract runvol'
```

Appendix E: Batch Processing Setup for Unix - using lp

Introduction to Batch Processing Setup for Unix using lp

Computron provides a facility within the `ctrone_install` script to configure a simple batch queue for background processing using the queuing system provided by Unix for spooling print jobs (`lp`). It's possible to extend the `lp` batch setup to consist of multiple `lp` batch queues for multiple jobs to run in the background simultaneously. It's also possible to use other queuing/batch systems for running background jobs. In order to do this, it's helpful if you understand the Computron batch processing interface programs.

Submittal of the Background Job

Background jobs are submitted into the batch queue via the `csbatch.sh` shell script stored in the `/.../exec/_ctrone_/` directory (where `/...` represents the file system in which the Computron application software resides). This script serves as the interface to the queuing system and can consist of a simple command to submit the job to a specific background queue or it may contain as much complicated logic as necessary to direct specific users and/or jobs to specific batch queues. It's up to the local system administrator to set up and maintain this script for the needs of the local site. Computron releases a template script called `csbatch.shd`, which is initially copied to `csbatch.sh` by the `ctrone_install` script during the initial installation of Computron software. This template consists of one executable line that issues an `lp` command.

The shell script must read the standard input for the Computron batch job control file (i.e., the Computron software merely opens a pipe to the `/.../exec/_ctrone_/csbatch.sh` command line). The control file is a specially formatted file that contains the Unix environment parameters, program data (`PUTPARM`), and Computron system variable (otherwise known as usage constants). The contents of the control file are submitted to the appropriate batch queue in the same manner that a user would submit a file to the printer for printed output. The batch queue must then read in the control file information and process the Computron background job accordingly. This section includes two examples of `csbatch.sh` shell script, one using the Unix standard `lp` command, and the second using the Unix batch command for submitting the background job.

Sample Shell Scripts

The following sample `csbatch.sh` scripts are used when submitting background jobs via the `lp` queuing system:

Bourne Shell

```
: Bourne shell
# csbatch.sh script to submit a Computron background to the UNIX
batch system.
#
/usr/bin/lp -dbbatchqueue -s -t CtroneBG
```

Korn Shell

```
# !/bin/ksh

# csbatch.sh script to submit a Computron background to the UNIX
batch system.

# Set temporary file names
# -----

tmpfile=/tmp/batch$$

# read batch job parameters from standard input
# -----

cat - >$tmpfile

# Submit the batch job via the UNIX batch facility
# -----

batch <<BATCH_INPUT
/ctron/exec/_ctron_/csbatrun $tmpfile
/bin/rm -f $tmpfile
BATCH_INPUT
```

Executing the Job in the Background

The batch or spool queue that is used to run the Computron batch job (which we will refer to as the batch-end) must be able to retrieve the Computron control file that was submitted via the `csbatch.sh` interface script. It must then be able to take that control file and execute the `/.../exec/_ctron_/csbatrun <spool-file-name>`). This requires that the `csbatrun` program be run as that Unix user ID. Some queuing systems can run the back-end as the user who submitted the spool file. Others run the back-end as user `lp` or some other user ID assigned by the system. For the latter case, Computron has provided a special program called `cssetuser`, which resides in the `.../exec/_ctron_` directory.

This program is run with `setuid` (set user ID) as root, which allows the program to assign the necessary user ID (and, if necessary, the group ID) for executing `csbatrun`. The `cssetuser` program checks to see that the real user ID of the process invoking it is the user that is designated as `lp` or `daemon` (depending on the Unix system). No other user is allowed to run this program for security reasons. The permissions on `cssetuser` should be set to: mode 4755, owner root):

```
-rwsr-xr-x  1 root  ctron cssetuser
```

Synopsis of the `csbatrun` Command

```
csbatrun [-n nice-value] <spool-file-name>
```

where:

-n nice-value

This is an optional parameter to assign a priority to the background job. If not specified, a default nice-value of 8 is used to lower the priority of the background job. See the Unix manual pages for the `nice` command for details about assigning priorities to processes.

<spool-file-name>

The name of the file that contains the control file information. This file may be a copy of the file that was actually submitted into the `csbatch.sh`, but which is kept in the queuing system's own spooling area.

Synopsis of the `cssetuser` Command

```
cssetuser username[.groupname] [-nice-value] <spool-file-name>
```

where:

username

This is the Unix username of the user who submitted the job. Each queuing system has its own method of tracking the user name, which can then be passed on to `cssetuser`.

groupname

This is an optional parameter that specifies the name of the Unix group that should be assigned to the background job. If not specified, the default group assignment is `ctron`.

-n nice-value

This is an optional parameter (see previous).

<spool-file-name>

(see previous)

By default, `cssetuser` uses the `/ctron/exec/_ctron_/csbatrun` path to start the background job—after setting the appropriate user and group ID. If this isn't the correct location, the environment parameter `CTRON_EXEC` must be set and exported prior to executing the `cssetuser` command to specify the directory path preceding `_ctron_/csbatrun` command (e.g., `export CTRON_EXEC=/ctronprod/exec`).

Sample lp Interface Script

The following sample interface script is used when submitting background jobs via the `lp` queuing system.

☞ *Note:* Normally, this script is released as `/.../misc/ctron.batch`. This does not apply, however, to AIX implementations.

```
# !/bin/sh

# lp interface for Computron batch processing

# $1 = request ID
# $2 = user ID
# $3 = title
# $4 = copies
# $5 = options

user="$2"

shift; shift; shift; shift; shift
```

```
files="$*"

chmod +r $files
# The environment parameter CTRON_EXEC may be set to allow
# cssetuser to execute the
# ../_ctron_/csbatrun program. The pathname for cssetuser should be
# changed to execute
# the corresponding "cssetuser" program if necessary.
#
CTRON_EXEC=/ctron/exec
export CTRON_EXEC

echo + /ctron/exec/_ctron_/cssetuser "$user" files >>
/tmp/lpbatch.out

/ctron/exec/_ctron_/cssetuser "$user" $files >> /tmp/lpbatch.out
2>&1

exit 0
```

Creating Unix Print Classes for Background Processing

This section describes the process for configuring Computron's background print classes, which can be accessed by multiple print queue initiators (or devices). These instructions include the Unix command-line syntax used for the generic SUN and HP systems, as well as instructions on using the GUI Smit interface to configure background processing for the AIX system.

Printer Class Setup on HP-UX

This section steps you through the process of creating a HP-Unix print class for background processing. For this example, the class name is `bckgrnd`, and the print queues are `batch1-3`. Also, the example assumes that the directory pathname for Computron is named `/ctron`.

Before You Begin

If the `batch1` queue already exists, you should disable and reject the queue before you start. To remove a pre-existing `batch1` queue, enter the following commands:

```
/usr/bin/disable batch1
/usr/lib/reject batch1
```

Creating a Printer Class with Multiple Batch Queue Initiators

1. Shutdown the Unix spooler.

```
/usr/bin/lpshut
```

2. Create multiple batch queue initiators.

```
/usr/lib/lpadmin -pbatch1 -v/dev/null -i/ctron/misc/ctron.batch
/usr/lib/lpadmin -pbatch2 -v/dev/null -i/ctron/misc/ctron.batch
/usr/lib/lpadmin -pbatch3 -v/dev/null -i/ctron/misc/ctron.batch
```

☞ *Note:* If the `ctron.batch` file is in a directory other than `/ctron`, then enter the new directory name after the `-i` option.

3. Allow print requests to enter the request directory for the newly added queues.

```
/usr/lib/accept batch1 batch2 batch3
```

4. Enable the newly added queues.

```
/usr/lib/enable batch1 batch2 batch3
```

5. Create a print class called `bckgnd`, consisting of print queues `batch1`, `batch2`, and `batch3`.

```
lpadmin -cbckgnd -p batch1
lpadmin -cbckgnd -p batch2
lpadmin -cbckgnd -p batch3
```

6. Allow print requests to enter the request directory for the newly added printer class.

```
/usr/lib/accept bckgnd
```

7. Restart the LP spooler.

```
/usr/lib/lpsched
```

8. Modify the batch script (`/ctron/exec/_ctron_/csbatch.sh`) and specify the new class name.

```
Default:           /usr/bin/lp -dbatch1 -s -t CtronBG
Using Class Name:  /usr/bin/lp -dbckgnd -s -t CtronBG
```

Custom Directories for Computron Executables

If Computron's software executables are not in the default `/ctron/exec` directory path, then make the following script changes.

Edit Interface Scripts

For each print queue that you added to the `bckgnd` class there is a corresponding interface script file in `/usr/spool/lp/interface` (i.e., `batch1`, `batch2`, and `batch3`). You must edit each of these scripts by replacing the default pathname `/ctron/exec` with your installation pathname.

Example (using default pathname):

```
echo + batch1:/ctron/exec/_ctron_/cssetuser "$user"
$files>>/tmp/lpbatch.out
/ctron/exec/_ctron_/cssetuser "$user" $files>>/tmp/lpbatch.out2>&1
```

Example (custom pathname = /mydir/exec):

```
echo + batch1:/mydir/exec/_ctron_/cssetuser "$user"
$files>>/tmp/lpbatch.out
/mydir/exec/_ctron_/cssetuser "$user" $files>>/tmp/lpbatch.out2>&1
```

Also, you must set the correct CTRON_EXEC pathname in the script to match your installation. In a designated section of the interface script you find a line that reads: CTRON_EXEC=/ctron/exec). Simply edit this line to match your installation pathname (e.g., CTRON_EXEC=/mydir/exec).

Printer Class Setup on Sun/Solaris

This section steps you through the process of creating a Solaris print class for background processing. For this example, the class name is `bckgrnd`, and the print queues are `batch1-3`. Also, the example assumes that the directory pathname for Computron is named `/ctron`.

Before You Begin

If the `batch1` queue already exists, you should disable and reject the queue before you start. To remove a pre-existing batch 1 queue, enter the following commands:

```
/usr/bin/disable batch1
/usr/lib/reject batch1
```

Also, if Computron's software executables are not in the default `/ctron/exec` directory path, then make the following changes to the shell script (`/.../misc/ctron.batch`), to reflect your custom pathname.

Example (using default pathname):

```
echo + batch1:/ctron/exec/_ctron_/cssetuser "$user"
$files>>/tmp/lpbatch.out

/ctron/exec/_ctron_/cssetuser "$user" $files>>/tmp/lpbatch.out2>&1
```

Example (custom pathname = /mydir/exec):

```
echo + batch1:/mydir/exec/_ctron_/cssetuser "$user"
$files>>/tmp/lpbatch.out

/mydir/exec/_ctron_/cssetuser "$user" $files>>/tmp/lpbatch.out2>&1
```

Also, you must set the correct CTRON_EXEC pathname in the script to match your installation. In a designated section of the interface script you find a line that reads: CTRON_EXEC=/ctron/exec). Simply edit this line to match your installation pathname (e.g., CTRON_EXEC=/mydir/exec).

Creating a Printer Class with Multiple Batch Queue Initiators

1. Shutdown the Unix spooler.

```
/usr/bin/lpshut
```

2. Create multiple batch queue initiators.

```
/usr/lib/lpadmin -pbatch1 -v/dev/null -i/ctron/misc/ctron.batch
/usr/lib/lpadmin -pbatch2 -v/dev/null -i/ctron/misc/ctron.batch
/usr/lib/lpadmin -pbatch3 -v/dev/null -i/ctron/misc/ctron.batch
```

☞ *Note:* If the `ctron.batch` file is in a directory other than `/ctron`, then enter the new directory name after the `-i` option.

3. Create a print class called `backgnd`, consisting of print queues `batch1`, `batch2`, and `batch3`.

```
lpadmin -cbckgnd -p batch1
lpadmin -cbckgnd -p batch2
lpadmin -cbckgnd -p batch3
```

4. Restart the LP spooler.

```
/usr/lib/lpsched
```

5. Allow print requests to enter the request directory for the newly added queues.

```
/usr/lib/accept batch1 batch2 batch3
```

6. Allow print requests to enter the request directory for the newly added printer class.

```
/usr/lib/accept bckgnd
```

7. Enable the newly added queues.

```
/usr/lib/enable batch1 batch2 batch3
```

8. Modify the batch script (`/ctron/exec/_ctron_/csbatch.sh`) and specify the new class name.

```
Default:          /usr/bin/lp -dbatch1 -s -t CtronBG
```

```
Using Class Name: /usr/bin/lp -dbckgnd -s -t CtronBG
```

Print Queue Setup on IBM AIX

This section steps you through the process of creating an AIX print class for background processing using the GUI SMIT interface. For the purpose of this example, the batch queue name is `batch` and the print queue devices (or initiators) are `init1-3`. Also, the example assumes that the directory pathname for Computron is named `/ctron`.

Before You Begin

By default, Computron uses the name, `batch`, as the background printer queue name. Therefore, if you want to change the batch queue name you must edit the `enq` statement in the `/ctron/exec/_ctron_/csbatch.sh` file.

Default entry for `csbatch.ch`:

```
user/bin/lp -d batch -s -t batch
```

☞ *Note:* At some installations the line may appear as follows: `enq -P batch -C -T Ctron_BG`

Creating a Print Queue with Multiple Print Queue Devices

1. From the SMIT main menu, select the `Print Spooling, Add A Print Queue, Other` option; then add the following entries. (If the Print queue already exists, skip to step #2.)

Name of queue to add	batch
Name of device to add	init1
Backend program pathname	/ctron/exec/_ctron_csbatrunc

2. Next, add the `init2` print queue device by selecting the Print Spooling, Add an Additional Printer to an Existing Print Queue, Other option; then add the following entries.

Name of device to add	init2
Queue to which to attach device	batch
Backend Output File pathname	FALSE
Backend program pathname	/ctron/exec/_ctron_csbatrunc

3. Finally, add the `init3` print queue device by reselecting the Print Spooling, Add an Additional Printer to an Existing Print Queue, Other option and then adding the following entries. (We recommend configuring no more than three queue devices for your background print queue.)

Name of device to add	init3
Queue to which to attach device	batch
Backend Output File pathname	FALSE
Backend program pathname	/ctron/exec/_ctron_csbatrunc

Appendix F: Batch Processing Setup for Unix - using csenq

Introduction to Batch Processing Setup for Unix using csenq

Computron has developed a feature rich alternative to the traditional standard of employing the Unix lp spooler for processing jobs in the background mode. This new feature is available only in the Computron Unix application software version 7.0 and up.

Benefits

- ◆ Greater flexibility
- ◆ Multiple background queues by default
- ◆ Reduced administration
- ◆ Easy modification of the configuration via the `csenqd.conf` file
- ◆ Store queue names from (A-Z) for background jobs
- ◆ Ability to see additional information on each of the queues and the jobs within each of the queues
- ◆ Ability to cancel background jobs, move jobs between queues, hold jobs for deferred execution, as well as job status logging.

Installation of csenq

The new background queue is normally setup during the installation process. However, modifications to the Computron environment may be accomplished by executing the program, `ctrone_install`, located in the Computron `root/EXEC/_ctrone_` directory. **Only** an individual authorized to act as root may execute the `ctrone_install` command. This individual must be fully cognizant of all ramifications of executing the `ctrone_install` command. To install the new service, select item 3 from the `ctrone_install` menu. A sample of this menu appears below:

This program may be used to:

- 1 - Load S/W and perform customization

This option is used to load the Computron Software from tape.

(Another menu will select the level of customization required.)

- 2 - Check and update files

This option performs a full check of the installation directory, sets file access permissions, renames default screen image, menu and epic directories, and modifies various files.

3 - Install Computron Batch (lp or csenq) queue

This option installs a batch queue for running background jobs.

4 - Install Computron crontab entry

This option installs a Computron crontab entry for nightly clean up of Computron worklibs, temporary files, etc.

(Another menu will prompt for cron job information.)

5 - Check and update Configuration

This option reviews your Computron configuration including your user file, LGMAP, Database maps, runtime executable, and other files and makes any adjustments that are required. This option should be run for both vision and database implementations.

Commands

The following csenq commands are available:

```
csenq -l [-v] [QueueNameList]
csenq -c job1 [job2 [job3 ...]]
csenq -h QueueNameList
csenq -r QueueNameList
csenq -a job1 [job2 [job3 ...]]
csenq -m QueueName job1 [job2 [job3 ...]]
```

Definition of command modifiers

The following are csenq command modifiers and definitions:

- ◆ -l list jobs in queue
- ◆ -c cancel a job or jobs
- ◆ -h hold a queue
- ◆ -r release a queue
- ◆ -v verbose output
- ◆ -a add incomplete job back to a queue
- ◆ -m move a job from one queue to another (does not apply jobs in progress)
- ◆ QueueName is a letter between A-Z
- ◆ QueueNameList is QueueNames concatenated (e.g. ABC)
- ◆ job# is numeric value of job (from csenq -l)

Sample output of the csenq command “csenq -lv”

```
$csenq -lv
CSENQD status:
  Log File =
  Logging Level = 1
  Group Init = ctron
  Admin List = root,ctronsys
Queue=A State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=B State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=C State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=D State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=E State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=F State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=G State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=H State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=I State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=J State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=K State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=L State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=M State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=N State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=O State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=P State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=Q State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=R State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=S State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=T State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=U State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=V State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=W State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=X State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=Y State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=Z State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
```

The CSENQD.CONF file

This file controls the variables that are passed to the `csenqd` daemon. The `csenqd` daemon is started automatically with the first submission of a job to background processing from within a Computron application. This file is owned by `root` and may only be modified by an individual with `root` authority. This individual must be fully cognizant of all ramifications in modifying the `csenqd.conf` file.

Permissions of file:

```
-rw----- 1 root   ctron   1713 Oct 04 08:07 csenqd.conf
```

Sample file contents

```
# csenqd.conf -- created Thu Oct  4 08:07:43 2001
# To provide csenqd with variables to work with

# LOGGING_LEVEL determines the verbosity of the logfile.
# LOGGING_LEVEL=0 (no log).
# LOGGING_LEVEL=1 (default; logs start,submitted,run & end.)
# LOGGING_LEVEL=2 (debug).
LOGGING_LEVEL=1
# LOGFILE=

# GROUP_INIT determines what setgroups() should be set to.
# GROUP_INIT=group1,group2,...
GROUP_INIT=ctrone

# ADMIN_LIST determines who have authority to execute special tasks.
# ADMIN_LIST=admin1,admin2,...
ADMIN_LIST=ctrone

# NICE sets the nice value of the individual queue.
# MAXJOBS sets the maximum jobs that can be run in the queue.
# STATE sets the queue to either: E = Enabled, H= Held.
QUEUE=A, NICE=5, MAXRUNJOBS=3, STATE=E
QUEUE=B, NICE=5, MAXRUNJOBS=3, STATE=E
QUEUE=C, NICE=5, MAXRUNJOBS=3, STATE=E
QUEUE=D, NICE=5, MAXRUNJOBS=3, STATE=E
```

QUEUE=E, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=F, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=G, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=H, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=I, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=J, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=K, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=L, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=M, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=N, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=O, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=P, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=Q, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=R, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=S, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=T, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=U, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=V, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=W, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=X, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=Y, NICE=5, MAXRUNJOBS=3, STATE=E
 QUEUE=Z, NICE=5, MAXRUNJOBS=3, STATE=E

Definition of entries in file

Entry	Explanation
LOGGING_LEVEL	Determines the verbosity of the log file LOGGING_LEVEL=0 (no log) LOGGING_LEVEL=1 (default; logs start, submitted, run & end) LOGGING_LEVEL=2 (debug) (If turned on – remember to turn it off when issue is resolved to prevent massive amounts of information being captured)
LOGFILE	Do not enter information here unless you desire to have the logfile be generated other than the default for logs (CTRON_ROOT/var)

Entry	Explanation
GROUP_INIT	Determines what <code>setgroups()</code> should be set to GROUP_INIT=group1,group2, If for some security reason you don't use the group <code>ctron</code> , substitute or add this other group here. Separate entries with commas – no spaces between entries
ADMIN_LIST	Determines who has authority to execute special tasks. ADMIN_LIST=admin1,admin2,... The defaults are <code>root</code> and <code>ctronsys</code> . If you wish to add additional members to this list do so with caution. Separate entries with commas – no spaces between entries. Members must be authorized to Unix and be in the Computron user file.
NICE	Sets the nice value of the individual queue. Don't change the default value unless you are familiar with the Unix <code>nice</code> command. Setting this parameter to (0) effectively defeats the background feature
MAXJOBS	Sets the maximum jobs that can be run in the queue. The default is (3) three. This may be compared to having 3 background queues using the Unix <code>lp</code> spooler. You may increase or decrease this parameter. Increasing the parameter may negatively impact overall performance of your Unix system and should only be changed if sufficient system resources are available. Setting the parameter to (0) effectively shuts down the queue
STATE	Sets the queue to either: E = Enabled or H= Held. You may set a specific queue to a held state. You might consider changing this parameter if you desire to have certain jobs not execute in an automatic fashion. You would be able to move a specific held job from the held queue to an enabled queue for processing. To have specific jobs run to a background queue other than the A queue, you must use a Character Cell session of Computron. Once a login to the Unix Computron Financial system is complete, you may have this session output background jobs to another background queue by depressing PF-25. Enter the queue change into the Queue Name field.

Sample DSI screen displayed by PF-25

```

#.                --- Set Defaults for User ---                DS1
Enter Indicated Data and Press (RETURN)
                    --- or Press (PF-16) to Exit

Data Base Map:    WHMAPSV                in  CSDBMAP                on  DATA

Default Company          Company scnio:          Batch Stream:
I/O Diagnostics    N  (Off/Summary/Full)
  
```

```
Work/temporary files:      WORKDIR  WK000030   WORKVOL  WRKVOL
User programs:            RUNDIR   CIGLLOAD   RUNVOL  EXEC
```

--- For Submitted Jobs ---

```
Run or Hold  R (R/H)          Priority:   0
Queue Name:                                     B
```

```
RUNTIME=/test_install/70gen/exec/_ctron_/wrunsyb,  REV=7.0.5.0
WISPGID=30,  PID=20914,  NODENAME=qars1,  OS=AIX(3 4)
```

Sample log file of job running in queue B

```
Queue=A State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5
Queue=B State=enabled Jobs=[Running=1, Waiting=0, Max=3] Nice=5
JOB #      USER  PROCESS-ID  START-DATE-TIME  STATUS
OPENFILE  OPENLIB  OPENVOL  PROGNAME  PROGLIB  PROGVOL  DESCRIPTION
=====
21      amf    12874  10/10/01 12:20:46  Running
WHMAPSV  CSDBMAP  DATA   RPT1000B CIGLLOAD EXEC    Trial Balance Report
```

Sample log file of job running in queue A

```
Queue=A State=enabled Jobs=[Running=3, Waiting=1, Max=3] Nice=5
JOB #      USER  PROCESS-ID  START-DATE-TIME  STATUS
OPENFILE  OPENLIB  OPENVOL  PROGNAME  PROGLIB  PROGVOL  DESCRIPTION
=====
22      amf          18952  10/10/01 13:24:53  Running
WHMAPSV  CSDBMAP  DATA   RPT1000B CIGLLOAD EXEC    Trial Balance Report
-----
23      amf          17904  10/10/01 13:24:57  Running
WHMAPSV  CSDBMAP  DATA   RPT1000B CIGLLOAD EXEC    Trial Balance Report
-----
24      amf          21952  10/10/01 13:24:59  Running
```

WHMAPSV CSDBMAP DATA RPT1000B CIGLLOAD EXEC Trial Balance Report

25 amf 10/10/01 13:25:01 Waiting

WHMAPSV CSDBMAP DATA RPT1000B CIGLLOAD EXEC Trial Balance Report

Queue=B State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5

Queue=C State=enabled Jobs=[Running=0, Waiting=0, Max=3] Nice=5

Appendix G – Configuration Change to Support a Terminal Server or Web Server

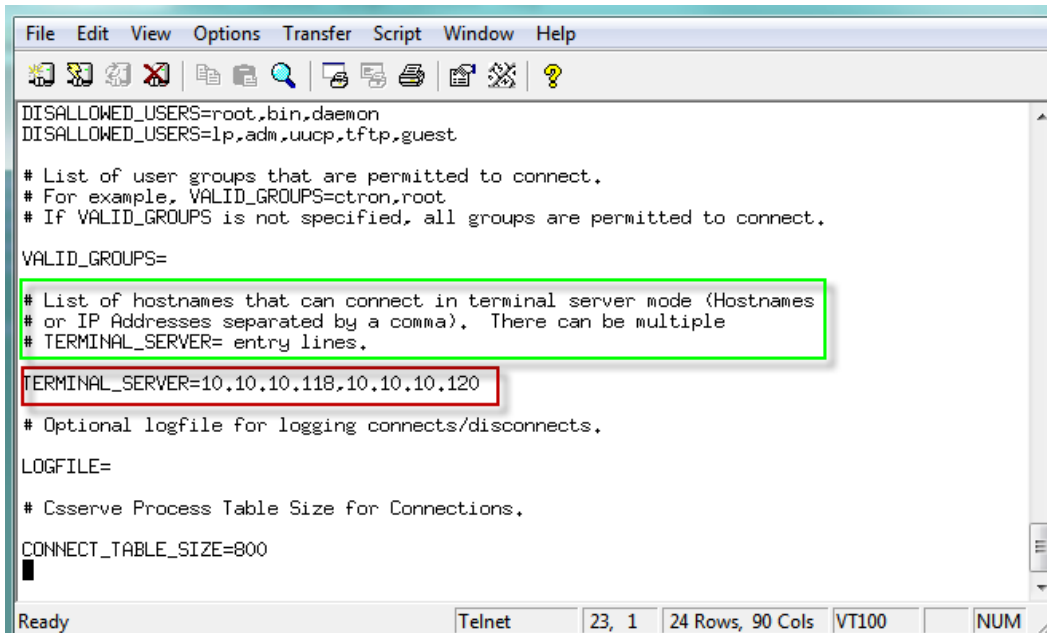
If this application server is used in conjunction with a Terminal Server running VB AUI, or a web server running WEBdesk or one or more e-Cellerators, a configuration change must be implemented. The application server needs to know that a web server or a terminal server is providing client connections, as opposed to a standard client/server VB AUI connection. Failure to implement this change results in dropped connections and various errors/hanging behavior, as the application server rejects multiple connections from the same IP address.

To implement this change, telnet to the application server, as root, and CD to `.../exec/_ctron_`. Locate the `csserve.conf` file and open it using `vi`.

Locate the line which begins with this string:

```
TERMINAL_SERVER=
```

An IP address must be entered for each Terminal or Web Server which will connect to this application server. Entries are delimited with the comma character and no spaces should be added. Please see the screenshot below:



```

File Edit View Options Transfer Script Window Help
DISALLOWED_USERS=root,bin,daemon
DISALLOWED_USERS=lp,adm,uucp,tftp,guest

# List of user groups that are permitted to connect.
# For example, VALID_GROUPS=ctron,root
# If VALID_GROUPS is not specified, all groups are permitted to connect.
VALID_GROUPS=

# List of hostnames that can connect in terminal server mode (Hostnames
# or IP Addresses separated by a comma). There can be multiple
# TERMINAL_SERVER= entry lines.
TERMINAL_SERVER=10.10.10.118,10.10.10.120

# Optional logfile for logging connects/disconnects.
LOGFILE=

# Cserve Process Table Size for Connections.
CONNECT_TABLE_SIZE=800

```

Ready Telnet 23, 1 24 Rows, 90 Cols VT100 NUM

Note the switch comments, highlighted in green, and the switch syntax, highlighted in red. Once the change has been made, save the changes and exit. In order for the csserve listener to read this configuration change, each listener port(s) must be killed and restarted, or a `csserve -reread` command must be executed, as root. In order to perform the `-reread`, please CD to `.../exec/_ctron_` and execute the following commands, as `ctronsys`:

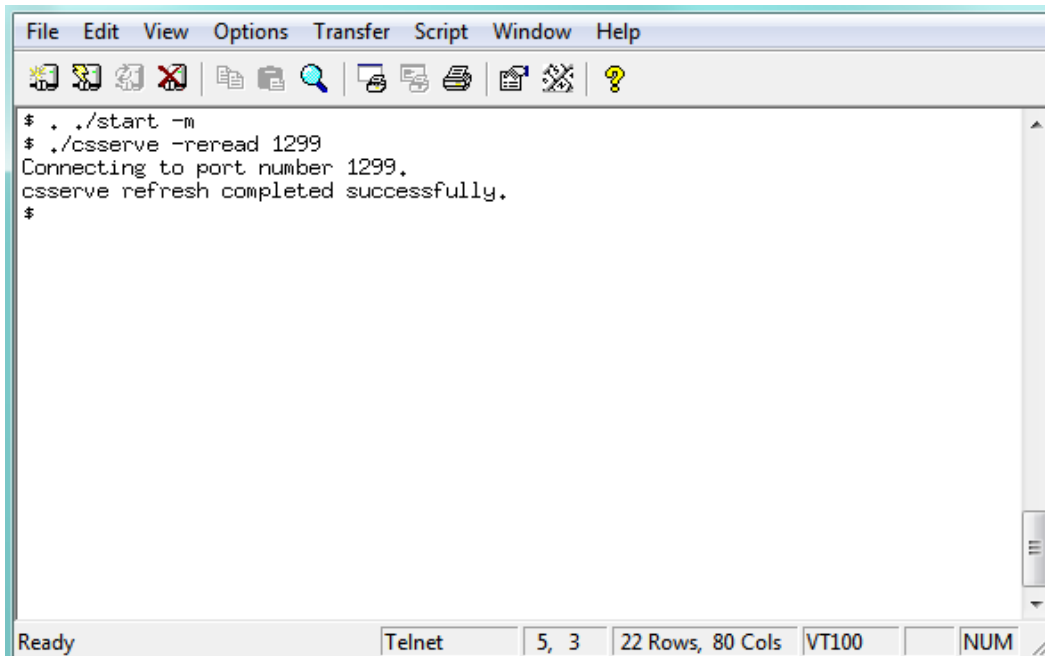
```
. ./start -m <ENTER> (this sets the Computron environment)
```

```
./csserve -reread {portnumber} <ENTER> (portnumber = the csserve listening port)
```

For example,

```
./csserve -reread 1299 <ENTER>
```

See the screenshot below:



Please note that it is necessary to `-reread` each csserve port which uses this file system.