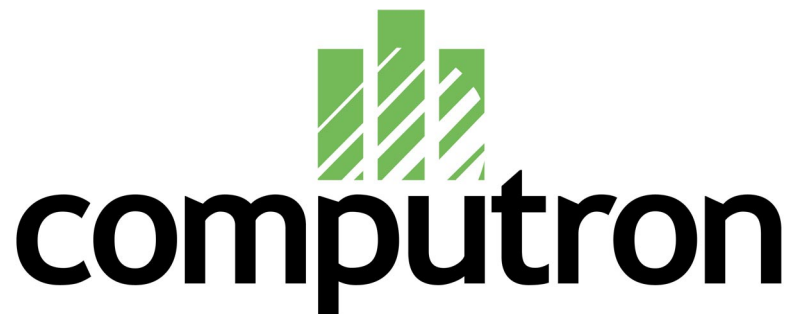


# **Microsoft Windows<sup>®</sup> System Administrators Guide**



Computron Software, LLC.  
301 Route 17 North  
Rutherford, NJ 07070  
(201) 372-6100  
[www.computronsoftware.com](http://www.computronsoftware.com)

## Revision History

First Edition		April 1998
Second Edition	Version 5.0	December 1998
Third Edition	Version 5.0	September 1999
Fourth Edition	Version 7.0	August 2002
Fifth Edition	Version 8w	June 2003
Sixth Edition	Version 10.0	November 2006
Seventh Edition	Version 11.0	December 2008
Eighth 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 1998 - 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](http://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.  
2<sup>nd</sup> 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 – Windows

## Table of Contents

### Preface

### Chapter 1—Universal Utilities

Overview: Setting Up 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 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 .....	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
Server Log Files .....	4.4-1
Clear Print Pass File .....	4.5-1
Clear Hold List Entries .....	4.6-1
Running Scripts on Windows .....	4.7-1
Script Logging on Windows .....	4.8-1
Accessing H-Key Registry Information .....	4.9-1

## Chapter 5—Computron Server Administration

Computron Server Administrator .....	5.1-1
Computron Batch Queuing Facility .....	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
Windows Processes – A Technical Overview .....	5.7-1
Configuring Computron’s Print Service .....	5.8-1
Installing a Hold Service .....	5.9-1
Installing a CSServe Service .....	5.10-1
Computron Process Monitor .....	5.11-1

## Chapter 6 – Database Utilities and Troubleshooting

RDBMS Utilities .....	6.1-1
RDBMS Loading and Unload Utilities .....	6.2-1
Miscellaneous Utilities .....	6.3-1
Database Logging (Debugging) .....	6.4-1
RDBMAP Format Maintenance .....	6.5-1
XMLMAP Format Maintenance .....	6.6-1
UEXTMAP Format Maintenance .....	6.7-1
Collecting Information Using the Computron Server Administrator .....	6.8-1
COBOL Debugging .....	6.9-1

## Appendices

Appendix A: Character Cell for Windows .....	A-1
Appendix B: UserExits .....	B-1
Appendix C: LGMAP File .....	C-1
Appendix D: Batch Queue Log File Messages .....	D-1
Appendix E: Backup and Recovery .....	E-1
Appendix F – Configuration Change to a Support Terminal or Web Server.....	F-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 Microsoft Windows® 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 Windows console, as well as a basic working knowledge of your Windows Server environment. In addition, the *WEBdesk Guide* is highly recommended if you have no previous experience with Computron software or Computron's graphical WEBdesk presentation.

The *WEBdesk Guide* 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: CSDISPLY – Computron's file display utility; 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 Windows, including the Computron Server Administrator, the Computron Process Monitor, the Computron Print Service System and the Computron Batch Queuing System.

## Chapter 6, Relational Database

This chapter provides detailed explanations of the Computron Database Utilities as well as various troubleshooting steps, including Backup and Recovery for NT, Script Logging on the NT Server and Accessing HKEY Registry Information.

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-<sup>Mark</sup> symbol beside the field name definition. The Q-<sup>Mark</sup> 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*.

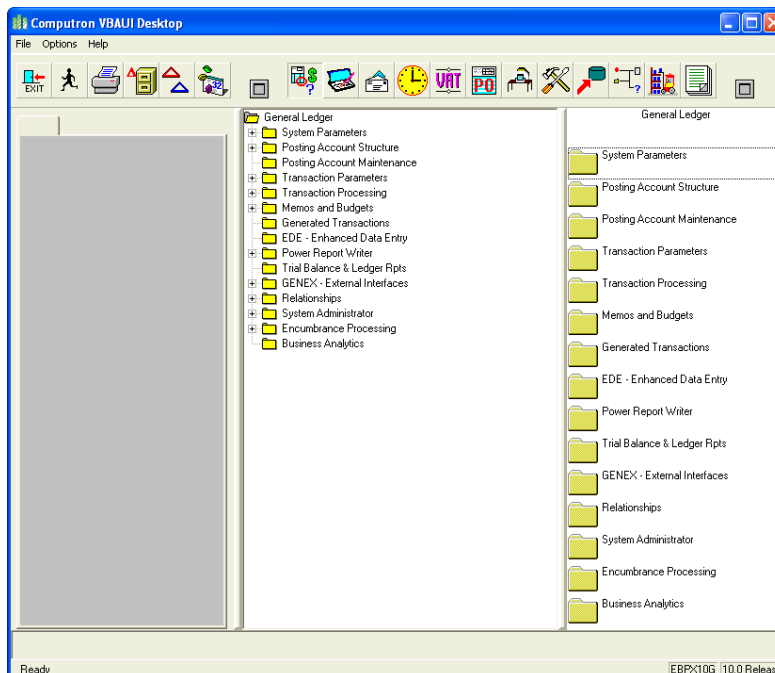
☞ *Note:* Soft Screen Maintenance may be included in the Universal Utilities menu for some installations.

## 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 by the General Ledger Master menu (GLMASTER).



**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

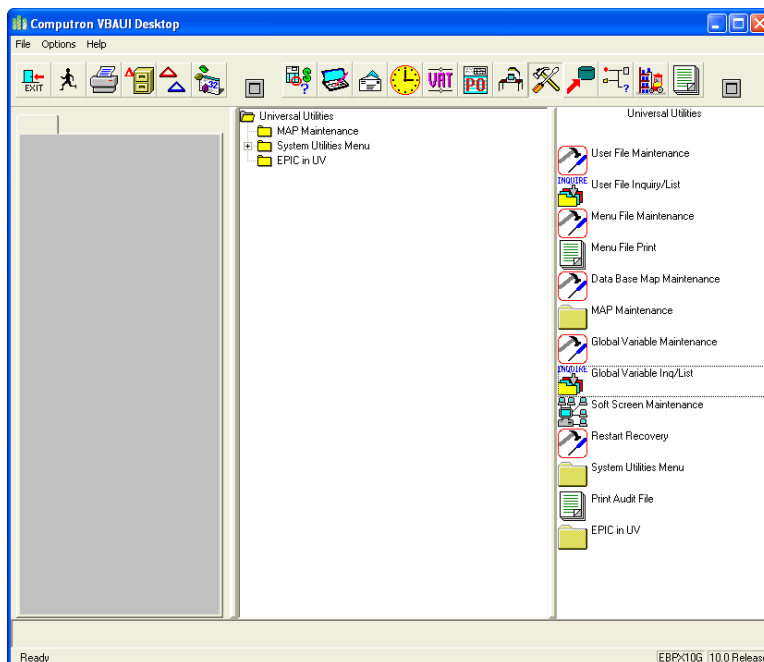
system) are grouped together. A list of available programs and their functions is available in the operation guide for each system.

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

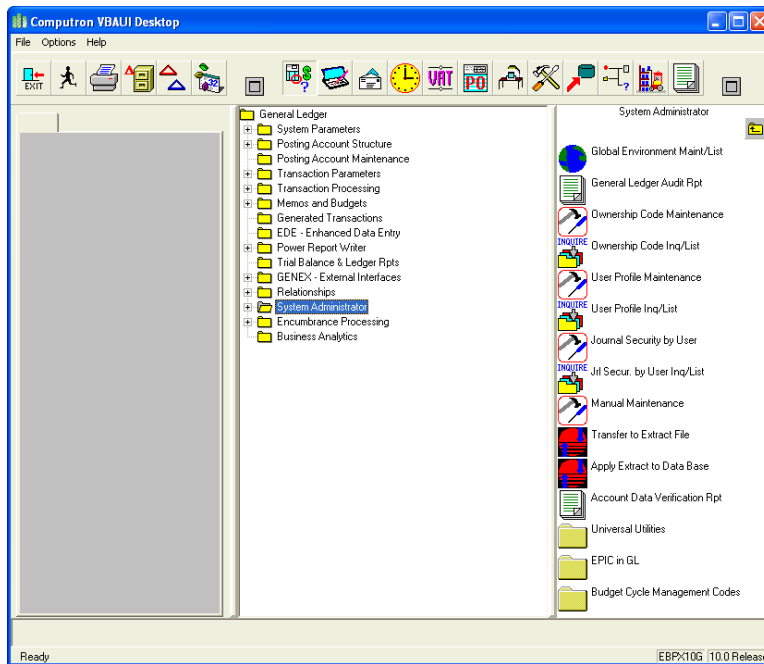
- ◆ Clicking a menu folder to display its contents in the Navigator list box.
- ◆ Clicking the plus symbol (+) next to a folder to display its submenus (if any) beneath; then clicking the submenu to display its contents in the Navigator list box. (Appears only with folders that contain submenus.)
- ◆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

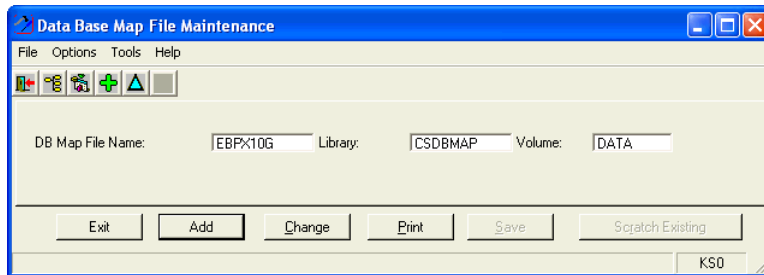
### *Key Windows (KS)*

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

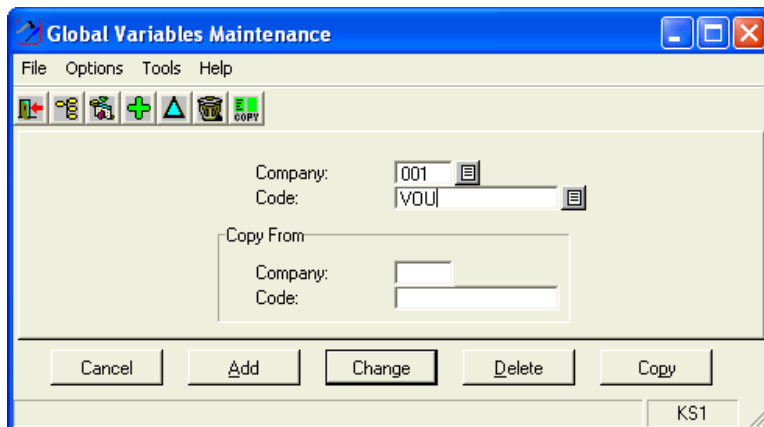
some Key windows, to select processing operations (e.g., add, change, delete, copy). Anytime you want to enter or change information for that data record, you must enter the key information on a Key window.

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**

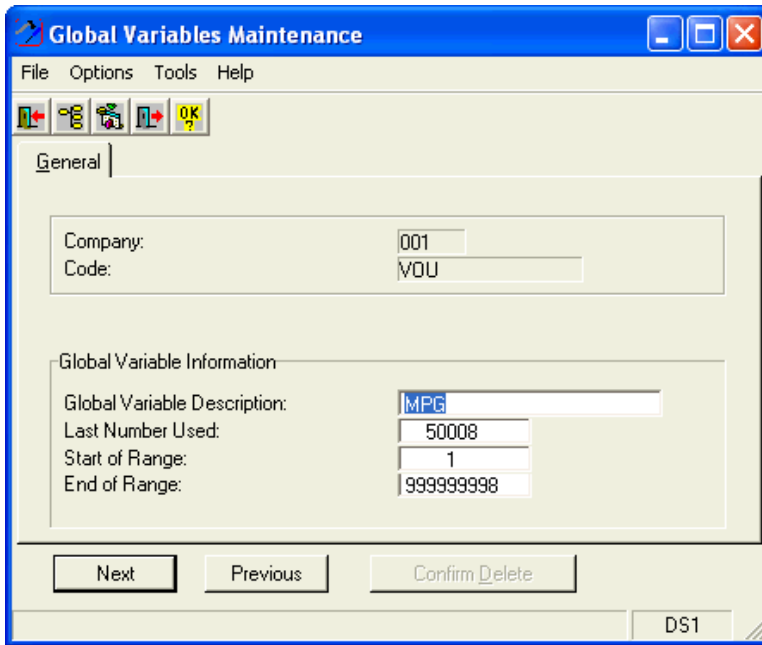


**KS1 Window - Key Window**

### *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.).

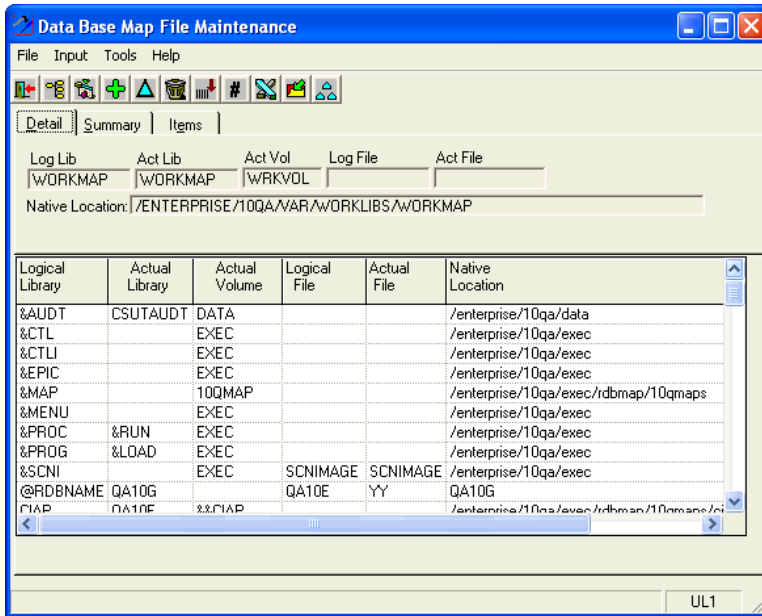


**DS1 Window - Data Window**

☞ *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)*

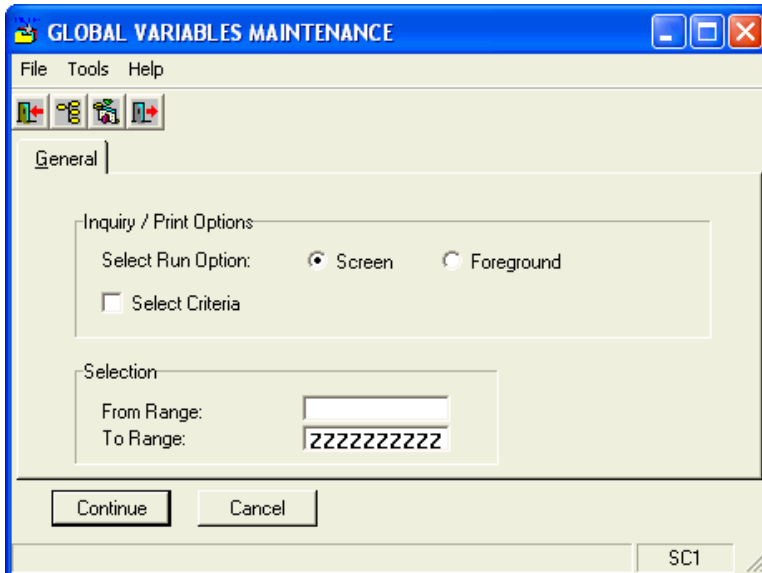
Universal Line Method windows are used to enter large volumes of data for functions like Data Base Map File Maintenance. This window allows you to enter multiple lines of data on a single window, speeding up the data entry process. Usually, this type of window is used to enter data that is repetitive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



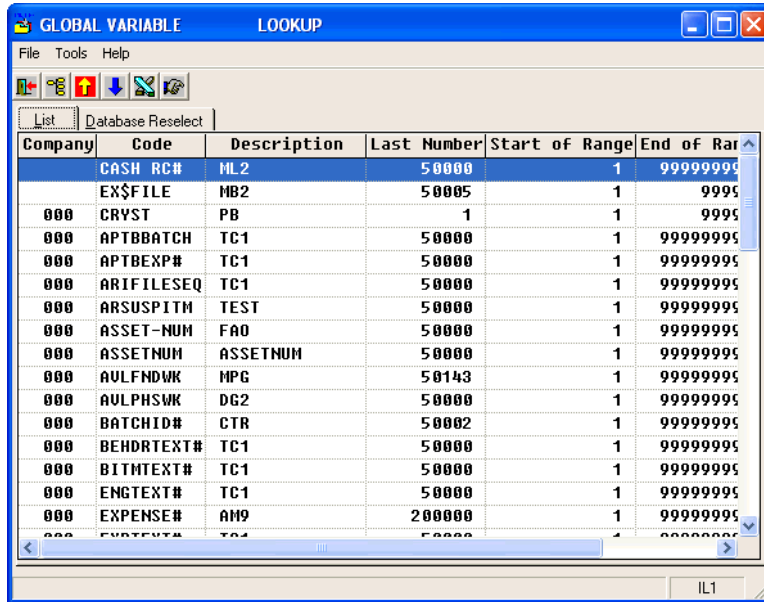
**SC1 Window - Selection Criteria Window**

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

This section is meant to provide a high-level overview of all the steps necessary to establish users for Computron software running in a Windows environment.

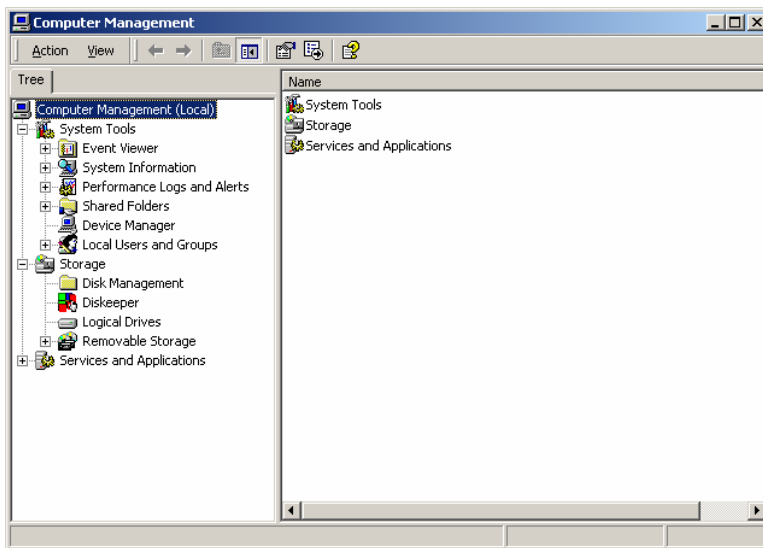
Before installing Computron software onto the Windows Server that will serve as the Computron application server, it is necessary to run Computer Management to create the following local User Objects:

- ◆ A Computron group (*ctron*).
- ◆ A Computron administrative user account (*ctronsys*), which is a member of both the *ctron* and Windows Administrator groups.

## Step 1: Create the Computron Group

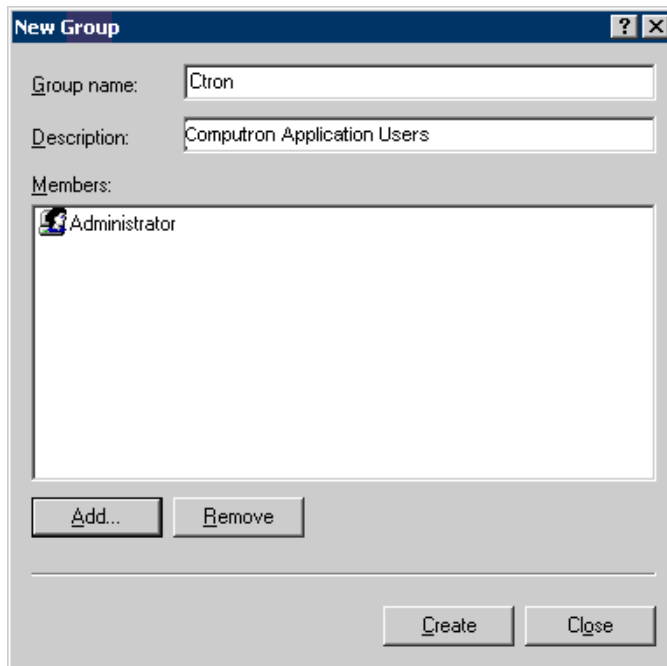
First, you must establish a Computron group (*ctron*) on the Computron application server.

1. Log onto the Windows Server console as an Administrator.
2. Click Start and then click the Administrative Tools menu.
3. Click Computer Management to display the Management Console.



**Computer Management Console Window**

4. Click Computer Management, System Tools, Local Users & Groups.
5. Right click Groups and select New Group.



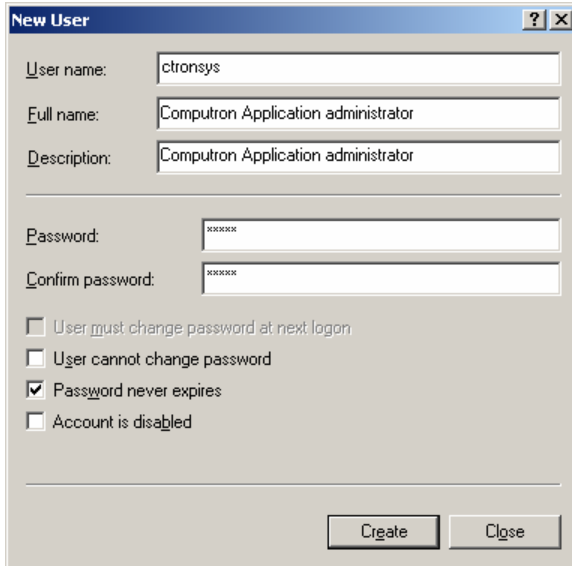
**Creating the Computron Group**

- a) In the Group Name box, enter Ctron.
  - b) In the Description box, enter Computron Application Users.
6. Click OK to create the group and return to the Computer Management window.

## Step 2: Add the Computron Administrator Account

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

1. From the Computer Management window, right click the User menu and then select New User to display the New User dialog box.

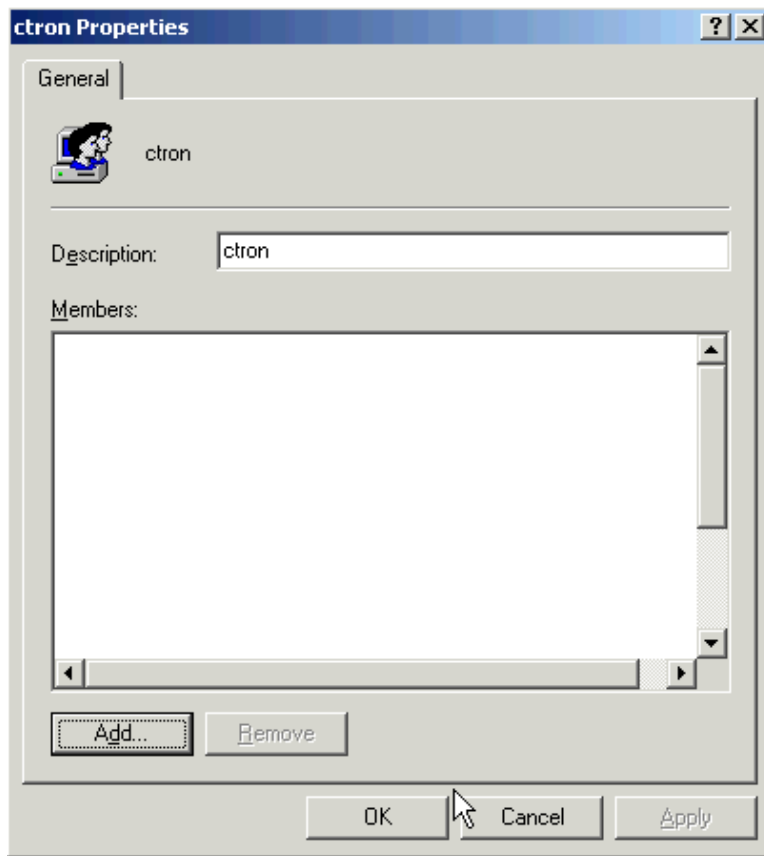


#### Creating the Computron Administrator Account

2. Enter the following user account information:
  - a. In the Username box, enter ctronsys.
  - b. In the Full Name box, enter Computron Application Administrator.
  - c. In the Description box, enter Computron Application Administrator.
  - d. In the Password and Confirm Password boxes, enter the desired password.
  - e. Clear the User Must Change Password at Next Logon check box.
  - f. Select the Password Never Expires check box to ensure that the Computron administrator is not inadvertently locked out of the system.

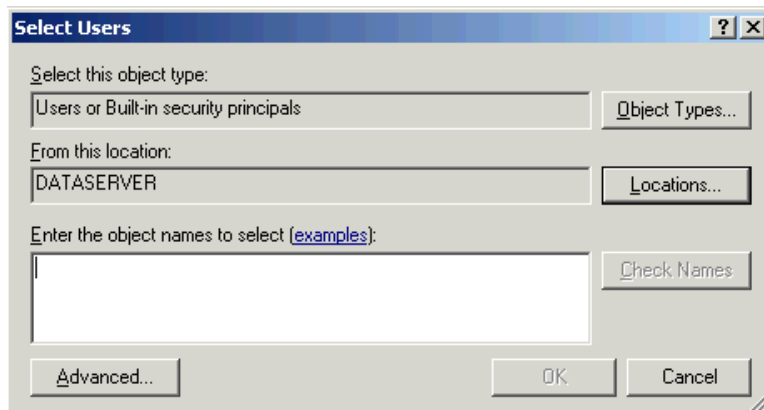
### Step 3: Adding the ctronsys User Account to Required Windows Groups

1. Click Groups from Computer Management.
2. Double Click the ctron Group.



**ctron Users Group**

3. Click the add button.

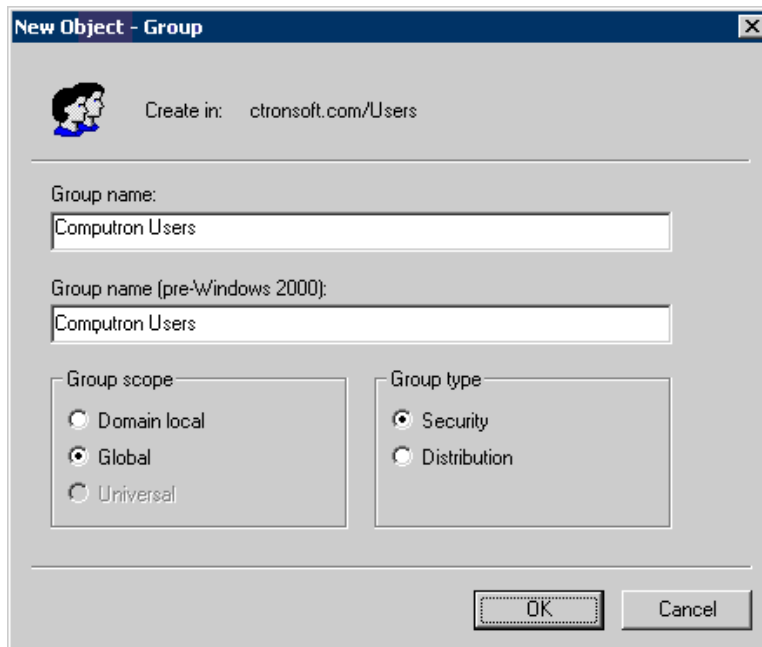


**Add ctronsys**

4. Enter "ctronsys" in the Enter the object names to select field. Make sure the value in the From this location field is set to the local machine.
5. Click OK and click OK again.
6. Repeat steps to add ctronsys to Administrators Group.

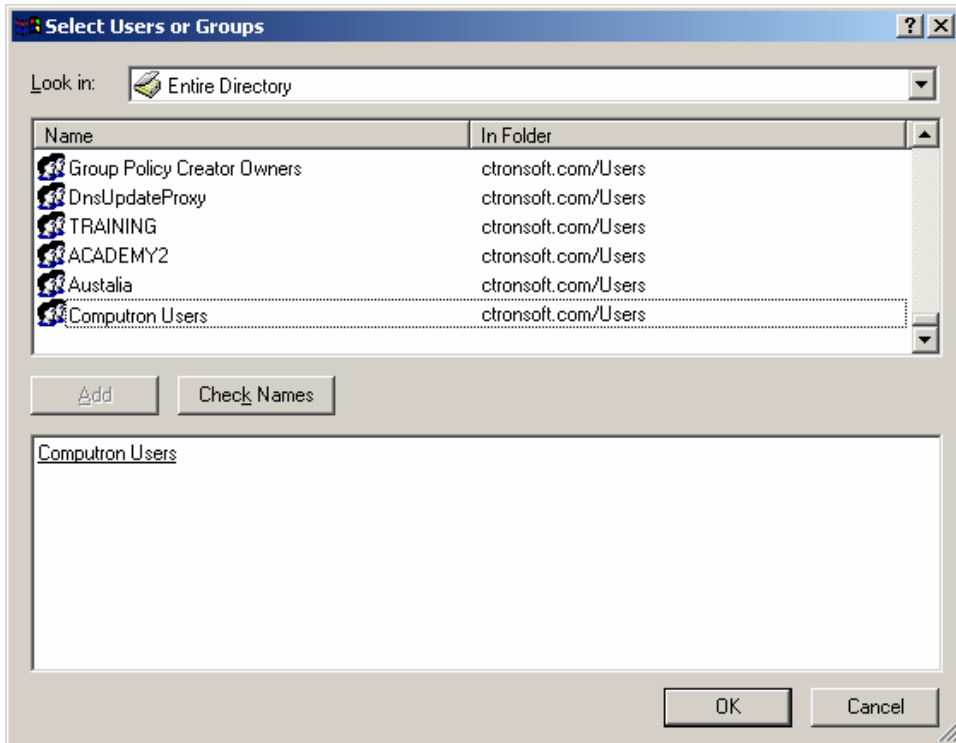
## Step 4: Adding Domain Users to Computron Application Server

- ☞ *Note:* You must be a Domain Administrator on a Domain Controller or a machine that has the Active Directory Users and Computers installed on it.
  - ☞ *Note:* Currently, Computron software recommends lowercase user IDs.
1. Create Domain Security Group. Click Start, Programs, Administration Tools, then click Active Directory Users and Computers.
  2. Right click on Users, select New, then select Group.



**New Group for Active Directory Users and Computers**

3. Enter a group name then select Global under group scope and select security under Group Type.
4. Click OK.
5. Find the new group just created and double click. Select the Members tab.
6. Click Add.

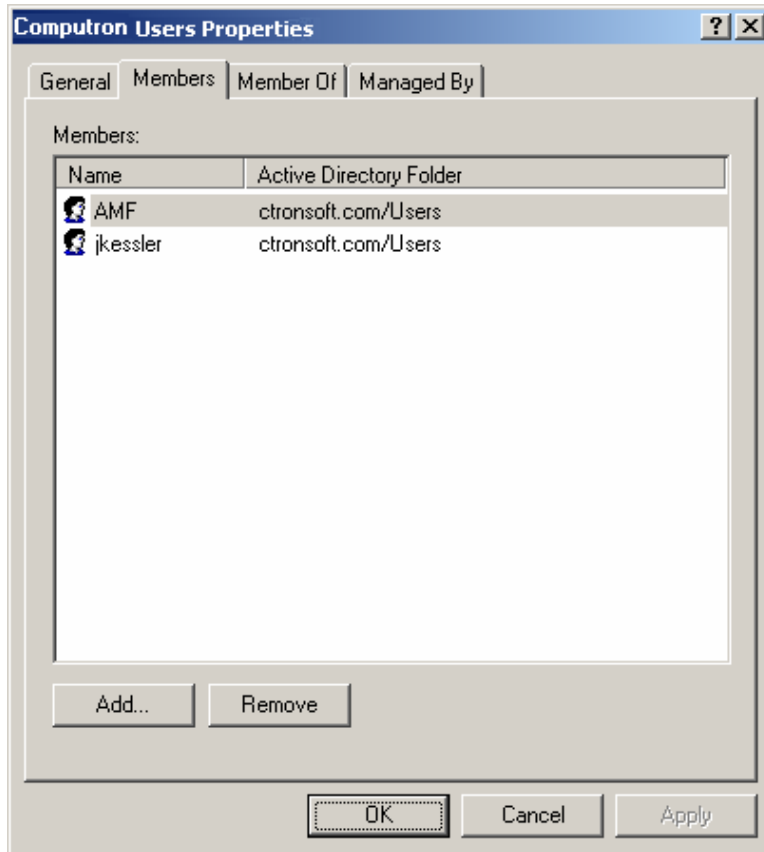


**Create domain group**

7. Select the Domain where the desired users exist.



8. Double click any user to add.



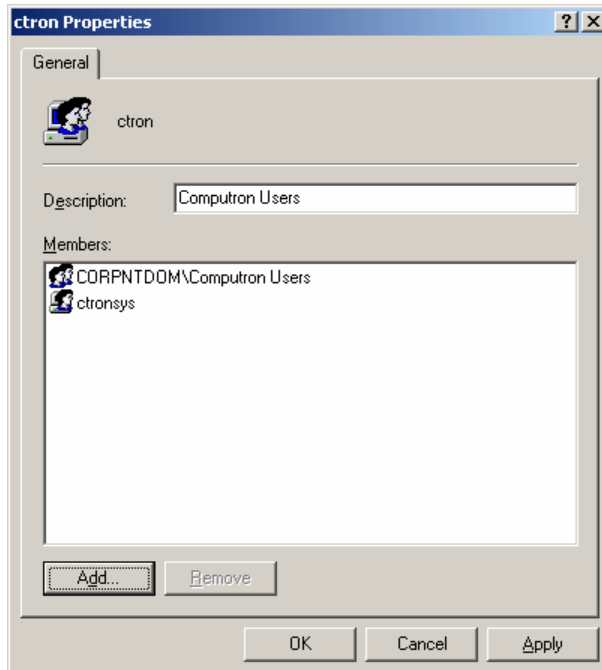
**Computron Users Properties**

9. Then click OK and OK again.

*Note:* You must be an Administrator and logged into the Computron application server to complete the following steps.

10. Click Start, Programs, click Administration Tools, click Computer Management.
11. Click Local Users and Groups.
12. Click Groups.
13. Double click the ctron group.
14. Select from the appropriate Domain.
15. Add previously created “Computron Users” domain group.

16. Click OK, then OK again.



Add Domain "Computron Users" group to local ctron group

## 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 Windows server.

## Install the Computron Software

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

## 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.

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 Windows user account, which is part of the Windows Security Access Manager (SAM) – by mapping the user key segment of the file to the Windows user account ID. The Windows user account indicates who can log onto the Window system. Computron's User file, in turn, indicates which of those individuals in the Windows user account can access Computron functions. Because the files are related in this manner, Computron currently maintains the following Windows user account requirements:

- ◆ Existing installations – have a unique five-character (5) limit – to accommodate print queue IDs.
- ◆ New installations – have a unique eight-character (8) limit.

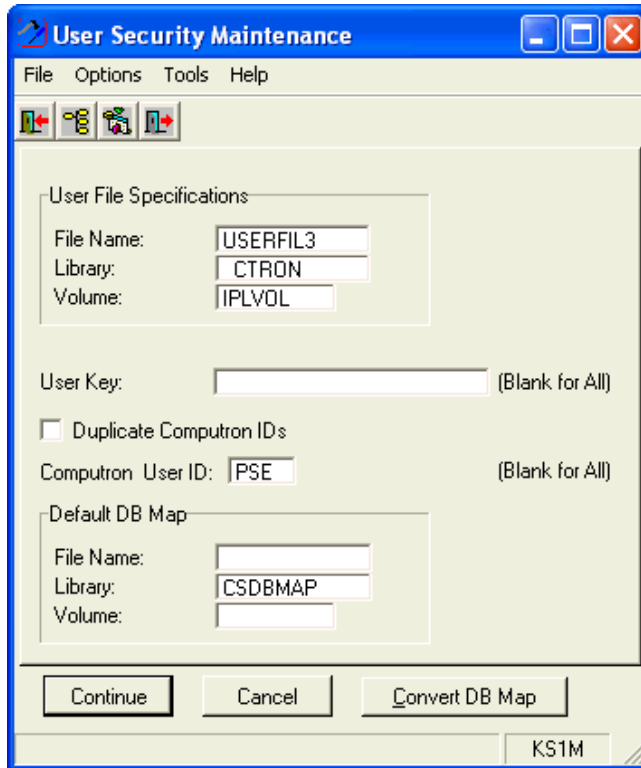
☞ *Note:* If a user without a User file record attempts to log on to Computron, the following message displays: “Invalid User ID or Password. Please try again.”

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 Windows 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 Windows 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 Windows 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.

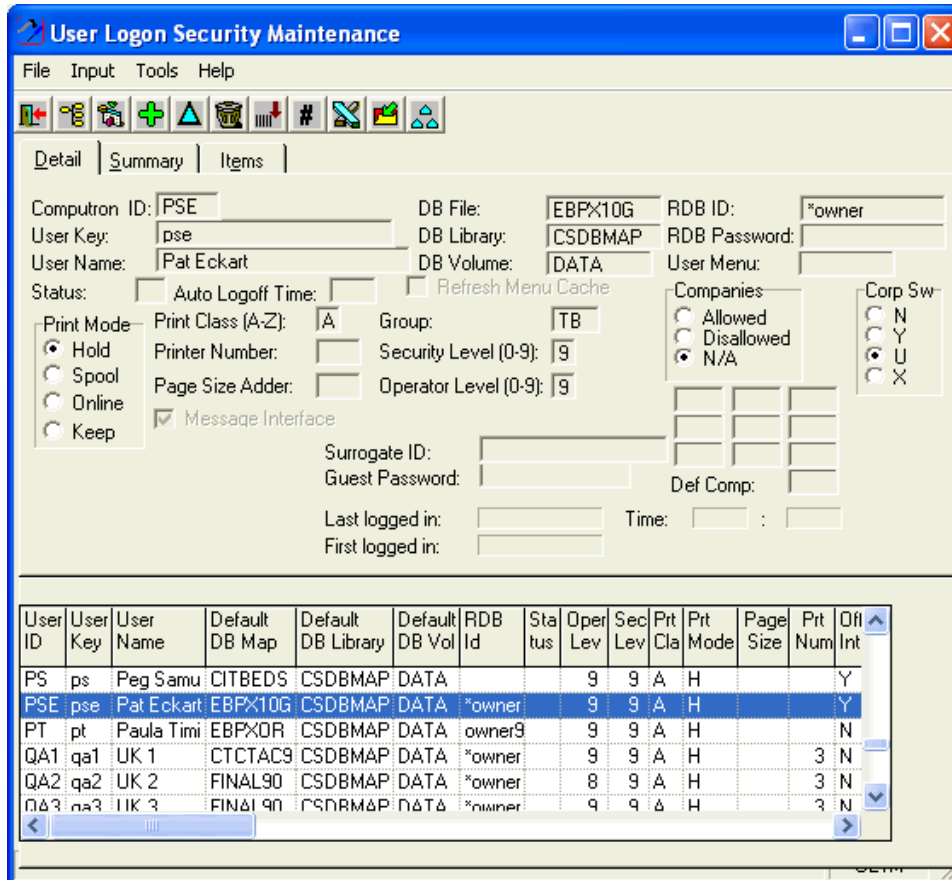
*Note:* Normally in relational database implementations, Computron systems are installed with the database map on the same volume as any parameter data files, so that it is not necessary to specify the volume in the database map for files/libraries on the same volume as the Database Map file

<b>Table 1.1-1</b>	
<b>User Security Maintenance Window (KS1) Options</b>	
<b>Button</b>	<b>Description</b>
<b>Continue</b>	Proceed to the User Logon Security Maintenance Window (UL1) where User file records are created, modified and deleted.
<b>Cancel</b>	Exit User File Maintenance.
<b>Convert DB Map</b>	<p>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.</p> <p><i>Note:</i> To process all records in the User file, leave the User ID field on the User Security Maintenance Window (KS1) blank.</p>

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: EBPX10G
- 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	Prt Cla	Prt Mode	Page Size	Prt Num	Ofi Int
PS	ps	Peg Samu	CITBEDS	CSDBMAP	DATA			9	9	A	H			Y
PSE	pse	Pat Eckart	EBPX10G	CSDBMAP	DATA	*owner		9	9	A	H			Y
PT	pt	Paula Timi	EBPXOR	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


This is the Windows 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 field 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.








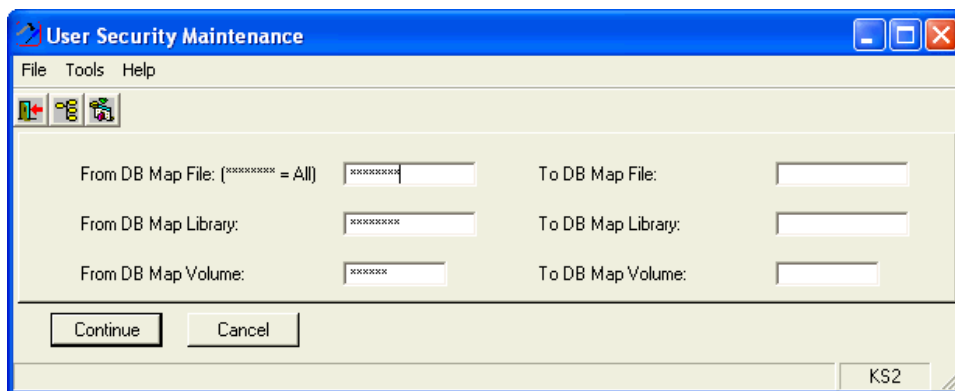
Table 1.1-2 Toolbar Options — User Logon Security Maintenance Window (DS1)		
Button	Function	Description
	Exit	Return to the User Security Maintenance Window (KS1) without saving any of the entries made.
	Add	Add a user record. This option presents the user currently highlighted in the list as a starting 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

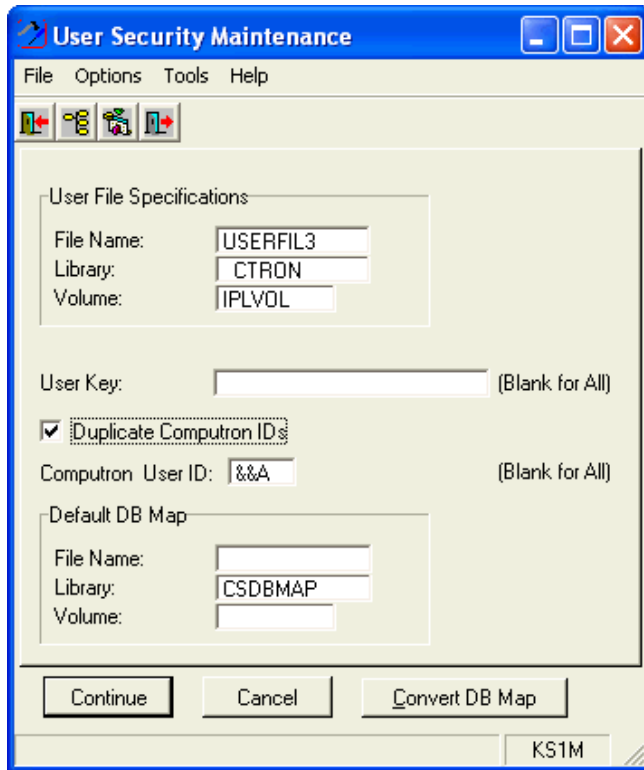
All RDBMS installations require a unique Computron user record, &&A, for each user that is an owner of Computron tables. Commonly referred to as amper-amper-A, the &&A record is used to connect to the database when either creating Computron tables or their stored procedures. Without it, various errors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

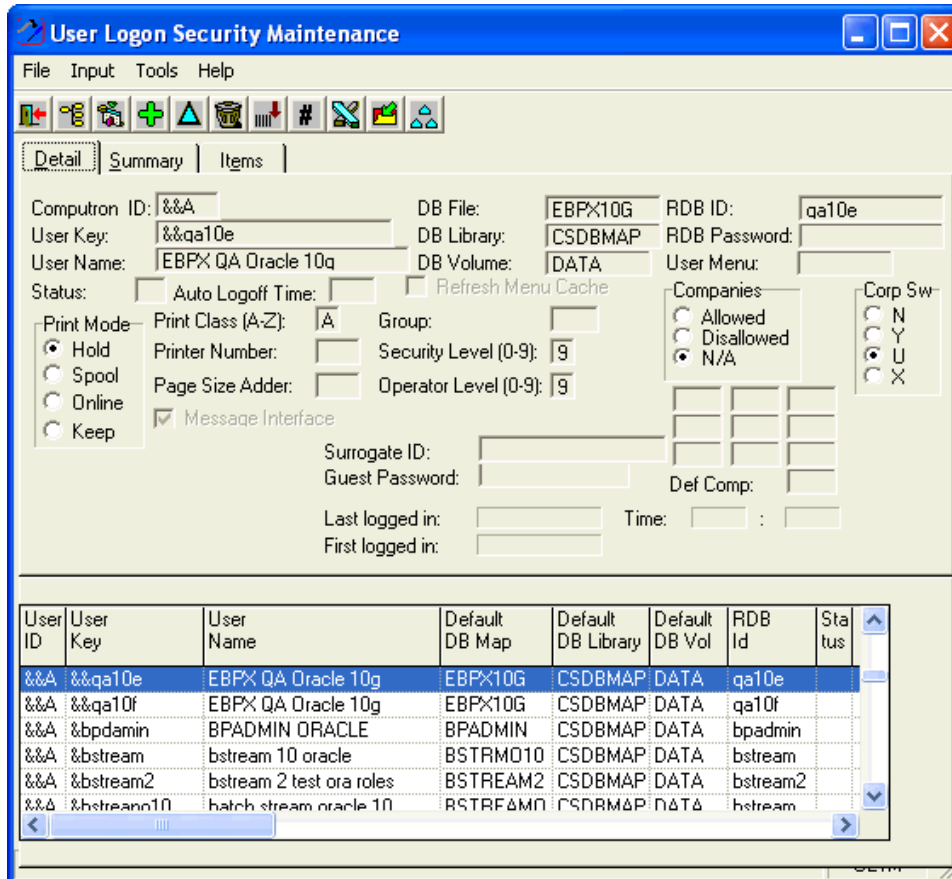
1. On User Security Window (KS1), select the Duplicate Computron IDs check box and enter &&A in the Computron User ID field.



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	batch stream oracle 10	BSTRF&M0	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.

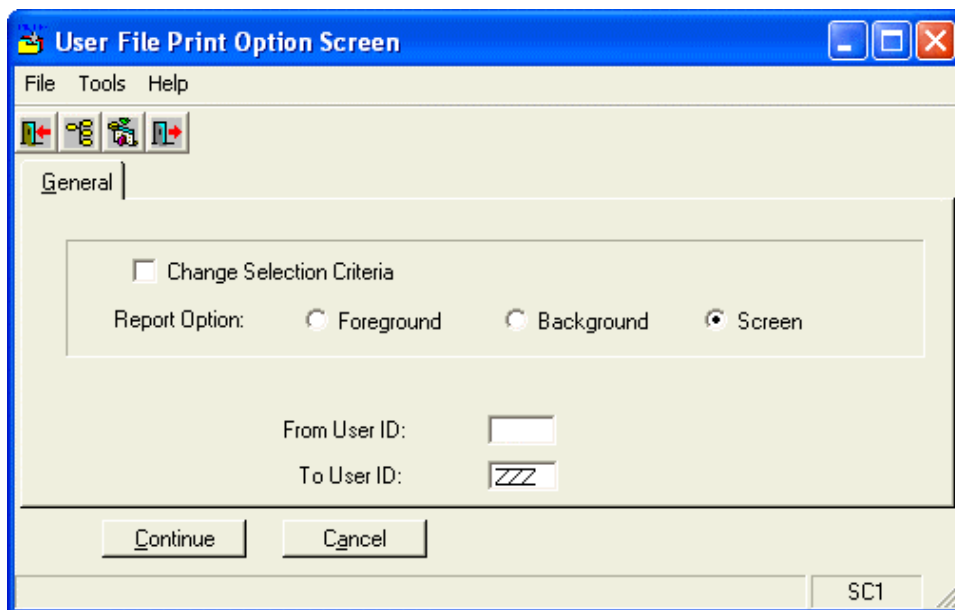
## 1.3 User File Inquiry/List

### *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.

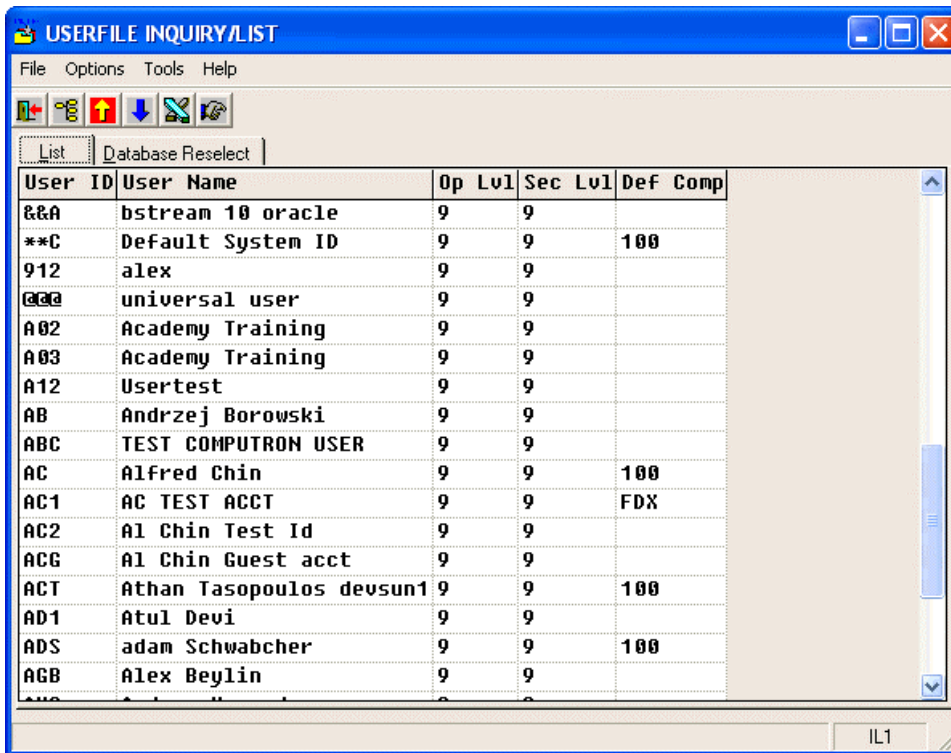
**To User ID**

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 buttons: "List" and "Database Reselect". The main area contains a table with the following data:

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	
000	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	

At the bottom right of the window, the text "IL1" is visible.

User File Inquiry/List Window (IL1)



# 1.4 Menu File Maintenance

## *Introduction to Menu File Maintenance*

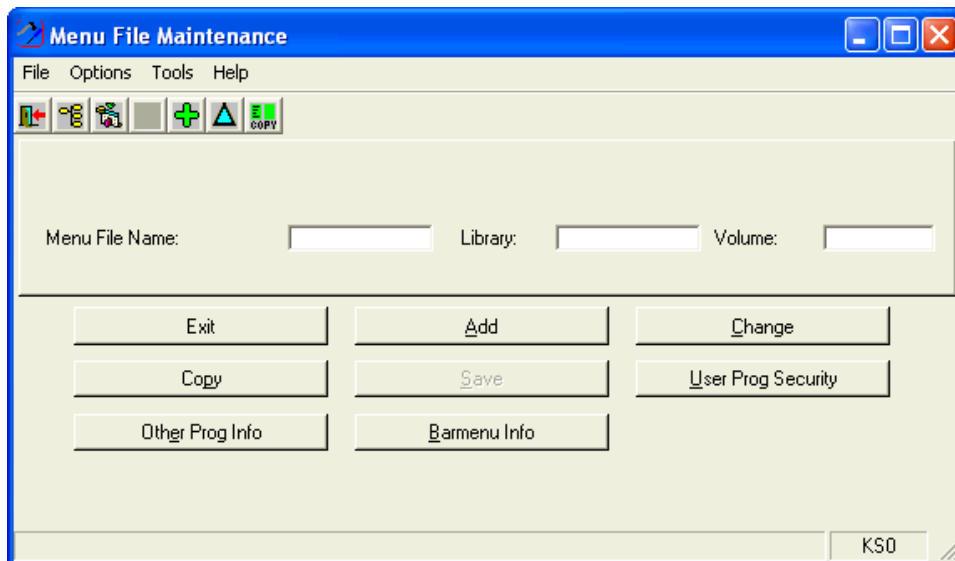
Computron menus are defined and maintained via the Menu File Maintenance (MENUENT2) function. The Menu File Definition windows allow you to change the format of any Computron menu. You can easily change a selection on the menu by editing the appropriate line item.

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.

<b>Table 1.4-1 Options on the Menu File Maintenance Window (KS0)</b>	
<b>Button</b>	<b>Function</b>
<b>Exit</b>	<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>
<b>Add</b>	<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>
<b>Change</b>	<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>
<b>Copy</b>	<p>Copy an existing Menu file.</p>
<b>Save</b>	<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="padding-left: 20px;">File Already Exists. Override To Scratch Existing File.</p> <p>Click the OK button to save the menu changes.</p>

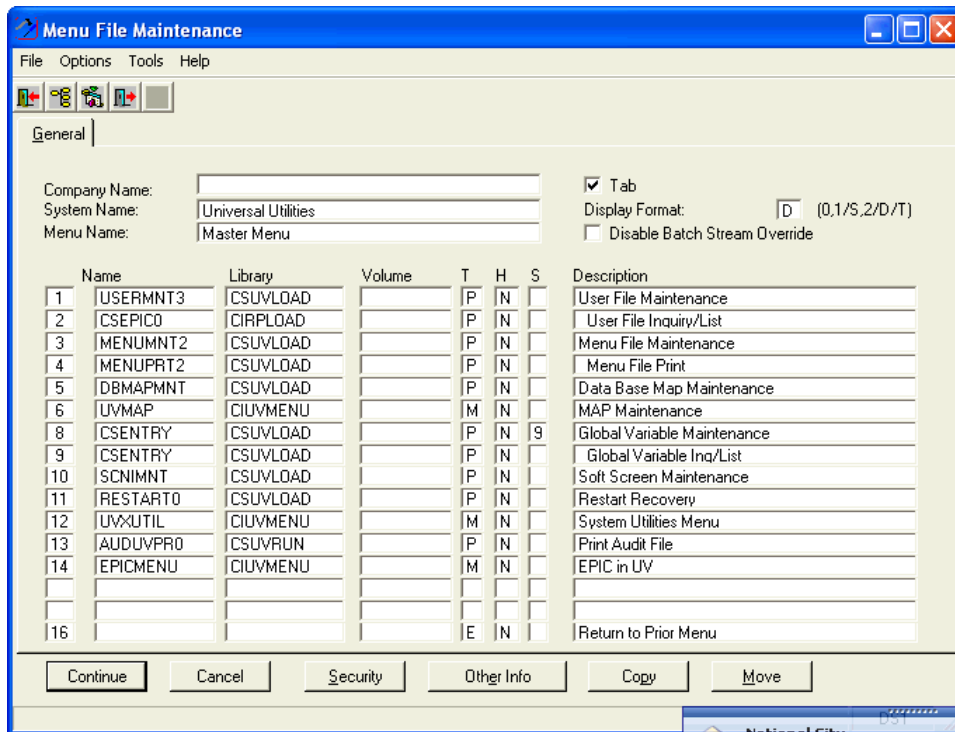


<b>Table 1.4-1 Options on the Menu File Maintenance Window (KS0)</b>	
<b>Button</b>	<b>Function</b>
<b>User Prog Security</b>	<p>Access the Security Maintenance Window — Menu File Security Maintenance Window (DS4). This window offers you a quick method of establishing security privileges for the Menu 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>
<b>Other Prog Info</b>	<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>
<b>Barmenu Info</b>	<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
<b>1</b>	<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
<b>S</b>	<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
<b>2</b>	<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
<b>D</b>	<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

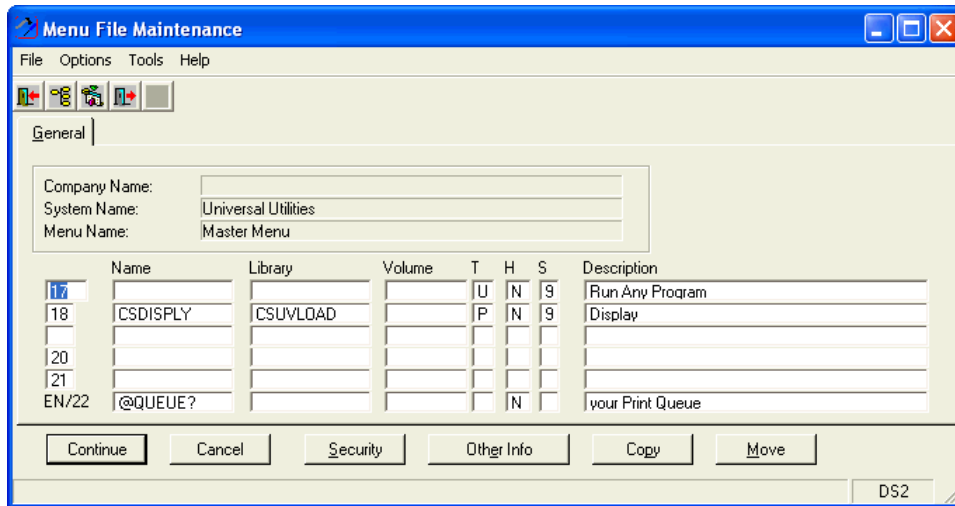
Option	Description	Max. Entries Per Column
T	<p>Creates a two-column menu that displays up to twenty selections: ten selections in the first column and up to ten selections in the second, until it encounters a selection type of E (Exit) or L (Logoff). For further details, refer to the T (Type) field 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

<b>Program</b>	System LOAD Library (e.g., CIGLLOAD for Computron's General Ledger system).
<b>Shell Script</b>	System RUN Library (e.g., CIAPRUN for Computron's Accounts Payable system).
<b>Menu</b>	System MENU Library.
<b>Switch To a New Database Map</b>	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:

<b>Special String</b>	<b>Substitution</b>
<b>&amp;IN</b>	Currently defined INVOL.
<b>&amp;OUT</b>	Currently defined OUTVOL.
<b>&amp;RUN</b>	Currently defined RUNVOL.
<b>&amp;WORK</b>	Currently defined WORKVOL.
<b>&amp;CTRON</b>	Currently defined CTRONVOL.
<b>&amp;DBMAP</b>	Currently defined DBMAPVOL.
<b>&amp;IPL</b>	Currently defined IPLVOL.
<b>&amp;SPOOL</b>	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 description. 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.

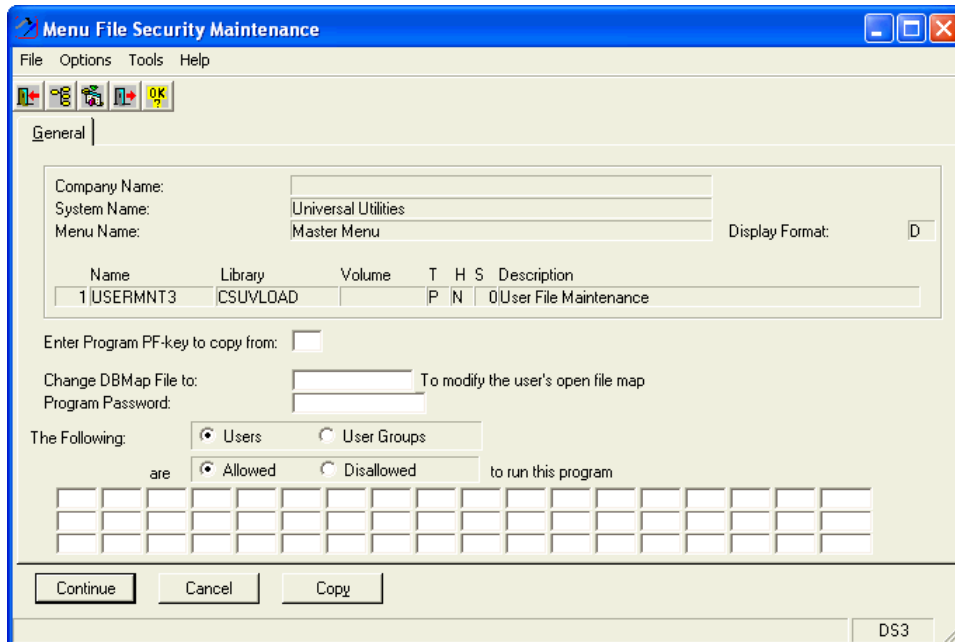
Once you have entered text on this window, you can easily manipulate the text using the Move and Copy buttons as described in the following table. Before clicking the Copy or Move buttons, be sure to examine the line item to which the text is being moved or copied. Any existing data on the line item 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
<b>Continue</b>	From DS1 — Proceed to the Menu File Definition Window (DS2). From DS2 — Proceed to the Menu File Maintenance Window (KS0).
<b>Cancel</b>	Return to the previous window.
<b>Security</b>	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.

<b>Other Info</b>	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.  <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.
<b>Copy</b>	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.
<b>Move</b>	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.

## 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.

<b>Table 1.4-4 Options on the Line Item Security Maintenance Window (DS3)</b>	
<b>Button</b>	<b>Function</b>
<b>Continue</b>	Proceed to Menu File Definition Window (DS1).
<b>Cancel</b>	Exit Line Item Security Maintenance Window (DS3) without retaining changes.
<b>Copy</b>	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:

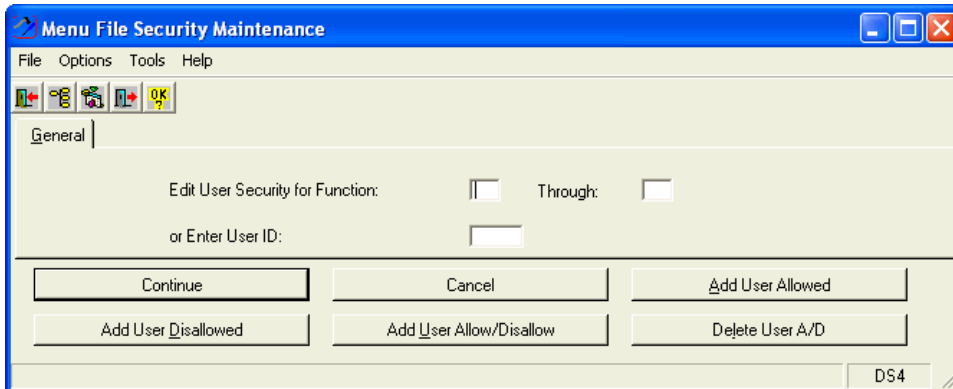
1. Access the Line Item Security Maintenance Window (DS3) and select:

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.

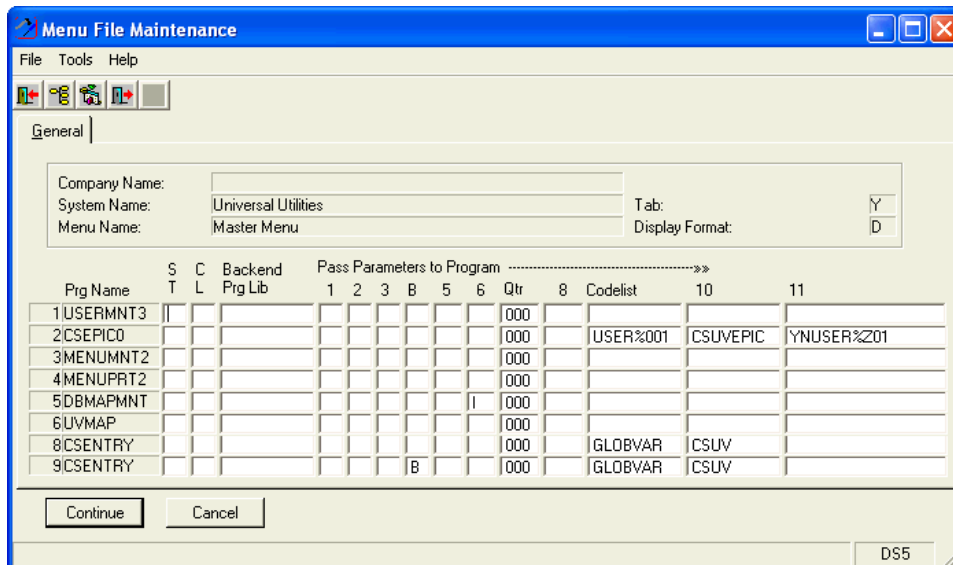
<b>Table 1.4-5 Options for the Security Maintenance Screen (DS4)</b>	
<b>Button</b>	<b>Function</b>
<b>Continue</b>	<p>Access the Line Item Security Maintenance Window (DS3) for the first line item in the range of function keys entered.</p> <p>Each time the Continue button is selected, the Line Item 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>
<b>Cancel</b>	<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>
<b>Add User Allowed</b>	<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>

<b>Table 1.4-5 Options for the Security Maintenance Screen (DS4)</b>	
<b>Button</b>	<b>Function</b>
<b>Add User Disallowed</b>	<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>
<b>Add User Allow / Disallow</b>	<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"> <li>◆ denied access to line items with the selection: "The Following Users Are Disallowed To Run This Program"</li> <li>◆ allowed access to those line items with the selection: "The Following Users Are Allowed To Run This Program"</li> </ul> <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>
<b>Delete User A/D</b>	<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-<sup>Mark</sup> 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.

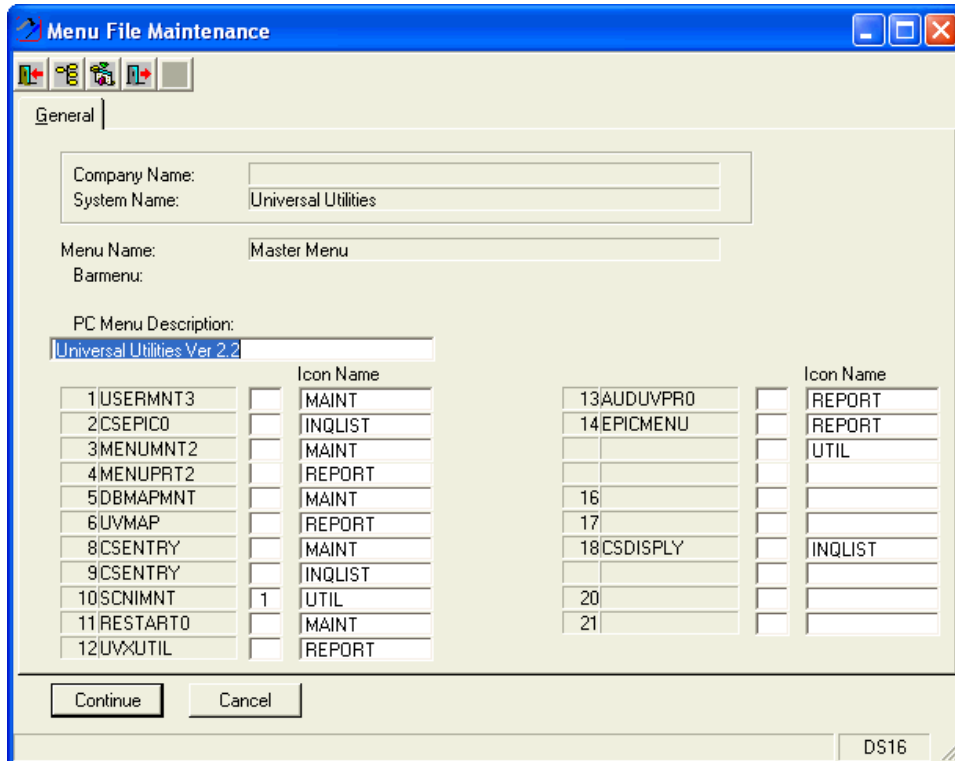
If no entry is made in positions three through ten, the system refers to the Codelist field and uses the soft screen program name as the report ID.

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 USERMNT3	MAINT	13 AUDUVPRO
2 CSEPIC0	INQLIST	14 EPICMENU
3 MENUmnt2	MAINT	
4 MENUprt2	REPORT	
5 DBMAPMNT	MAINT	16
6 UVMAP	REPORT	17
8 CSENTRY	MAINT	18 CSDISPLY
9 CSENTRY	INQLIST	
10 SCNIMNT	1 UTIL	20
11 RESTART0	MAINT	21
12 UVXUTIL	REPORT	

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.

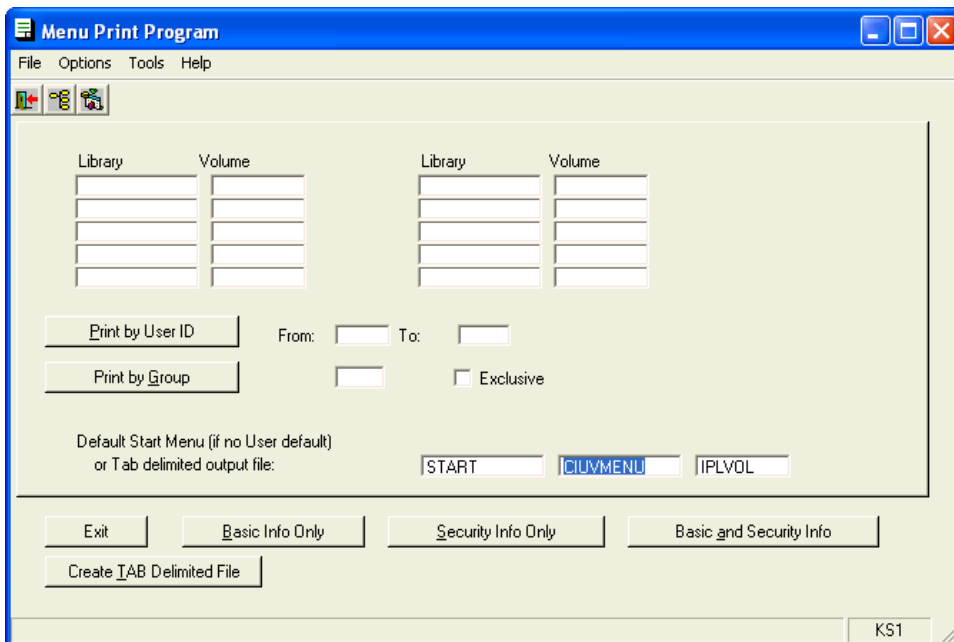
☞ *Note:* When replacing icons, make sure to copy your replacement icon files into the appropriate location before accessing this function.

## 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.

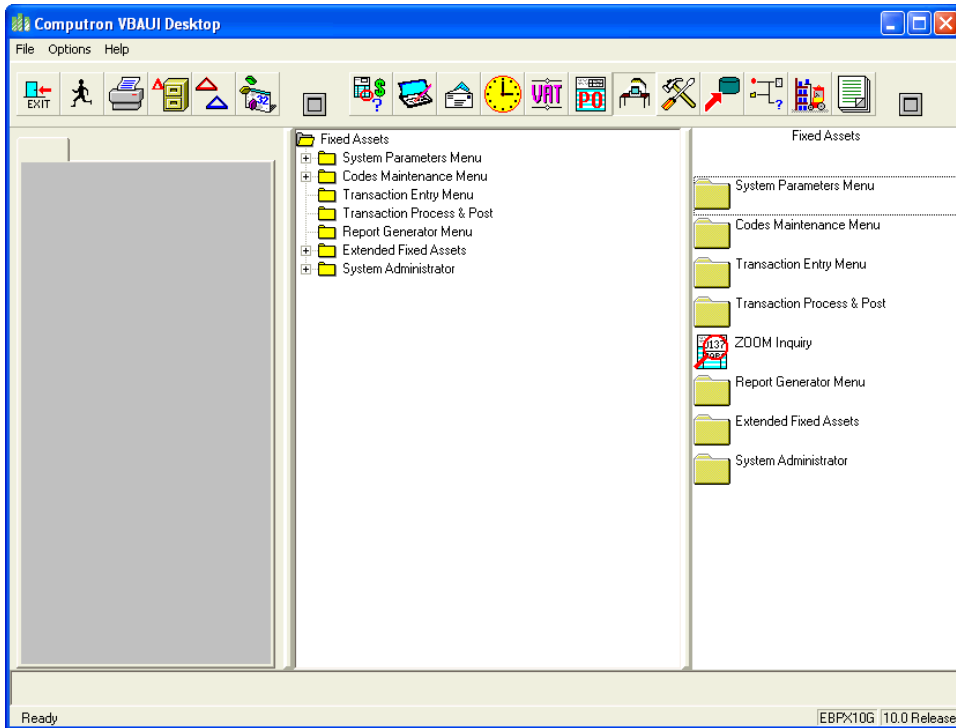
Before beginning this process, you should print a listing of the menus as they have been defined by running the Menu File Print function. See Section 1.5, “Menu File Print,” for additional information. This listing can be used as an audit trail, in case you inadvertently make unwanted changes to a menu 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
<b>PF4</b>	Transaction Process & Post	6
<b>PF5</b>	Zoom	1
<b>PF6</b>	Report Generator Menu	1
<b>PF7</b>	Extended Fixed Assets	7
<b>PF8</b>	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.

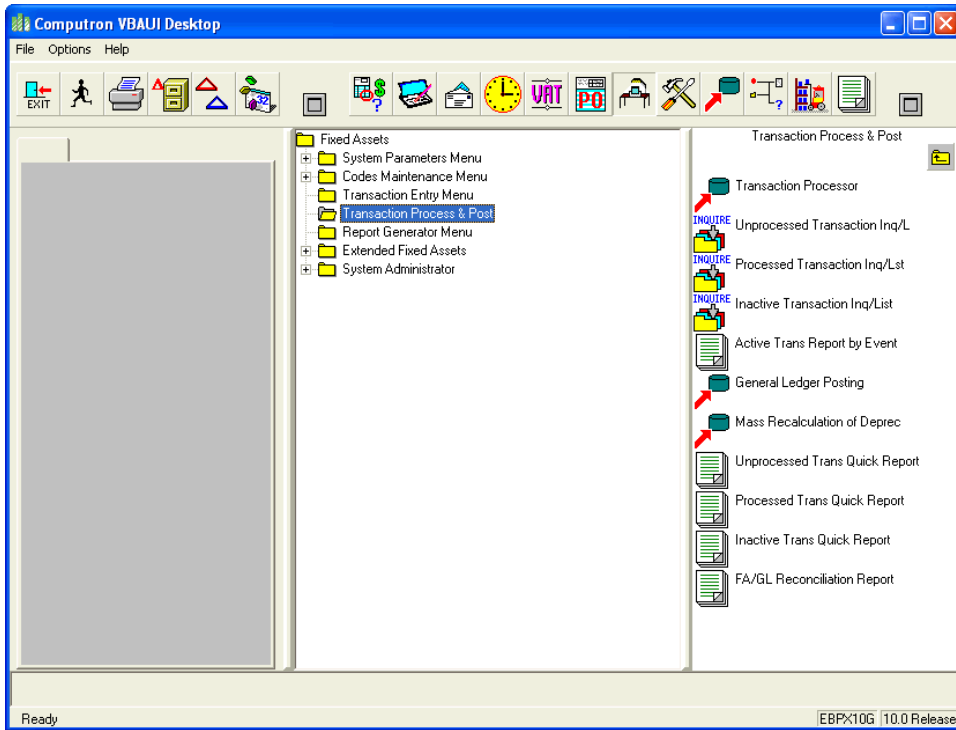
This means that the user must have an equal or higher security level (established for each user in Computron's User File Maintenance function) in order to select the option. Those menu selections to which the user does not have access are not displayed on the menu window.

☞ *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).

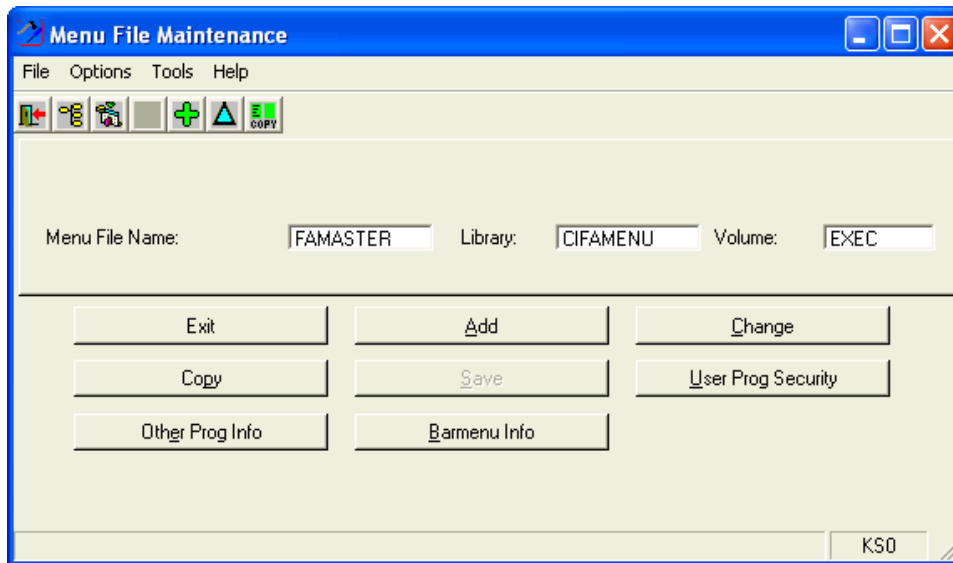
Computron Report ID	Tailored Report ID
TRAN%001	TRANC001
TRAN%002	TRANC002
TRAN%004	TRANC004

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

## Solution

### *Solution — Establish Security*

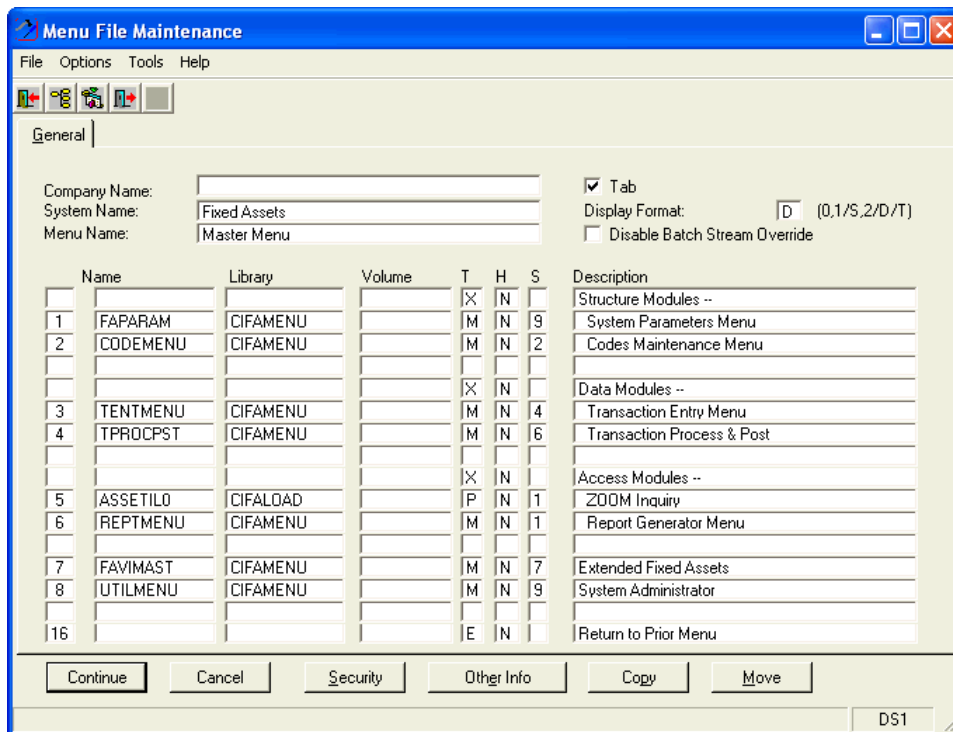
Access the Menu File Maintenance function and enter the Menu File Name/Library/Volume as shown here:



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:



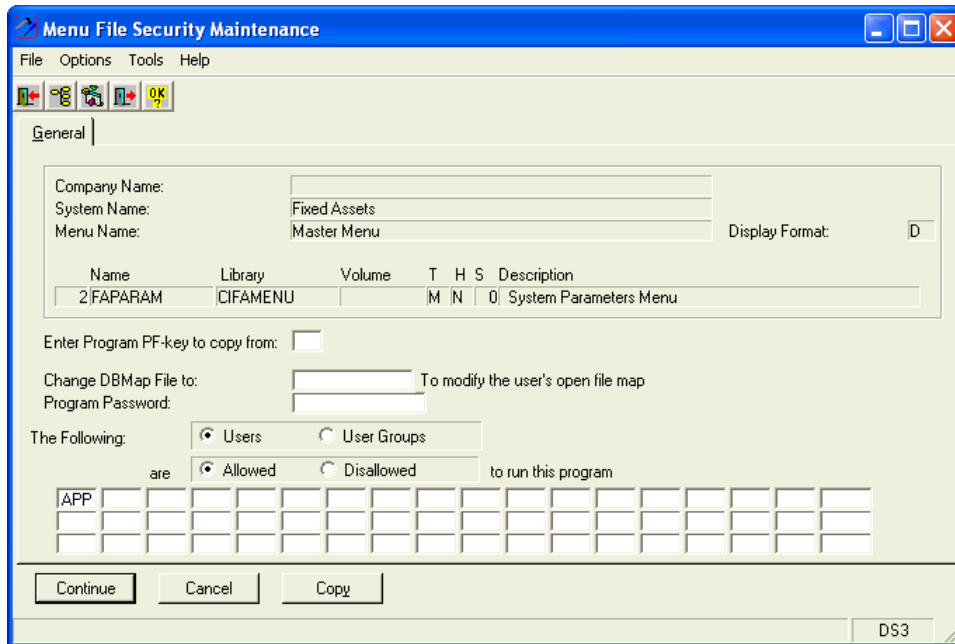
Menu File Maintenance Window (DS1) for FAMASTER

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

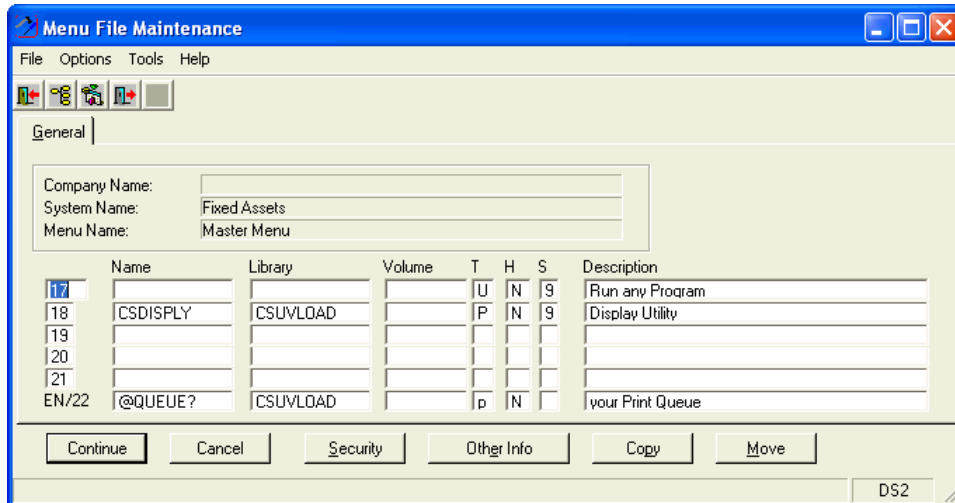
Only the user ID indicated above should be allowed access to the particular menu selection. In order to achieve this, position the cursor on the line for the System Parameters menu selection and click the Security button. The following DS3 window displays:



**Menu File Maintenance Window (DS3) for FAMASTER**

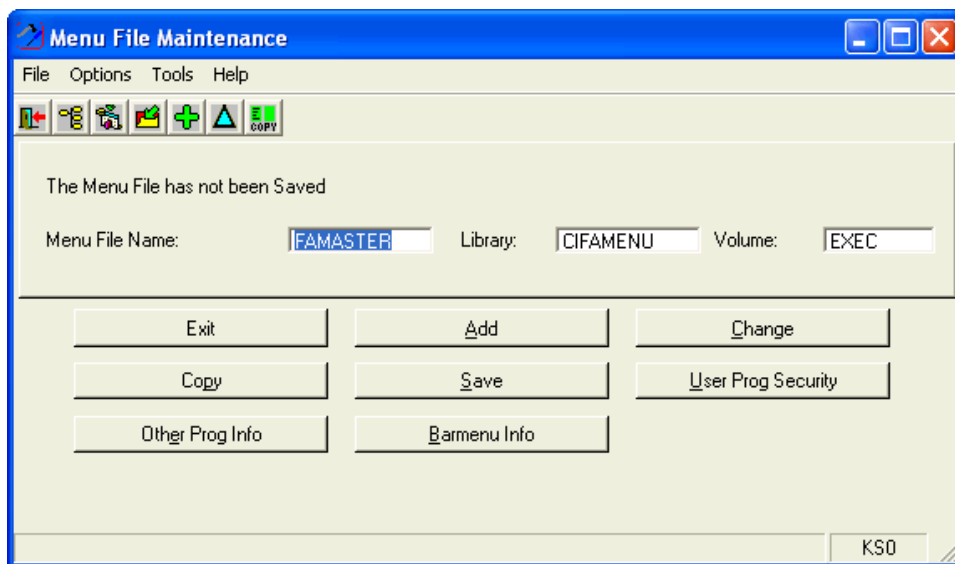
Click Continue to return to the original Menu File Definition window. Now modify the System Administrator menu selection in the same way.

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*

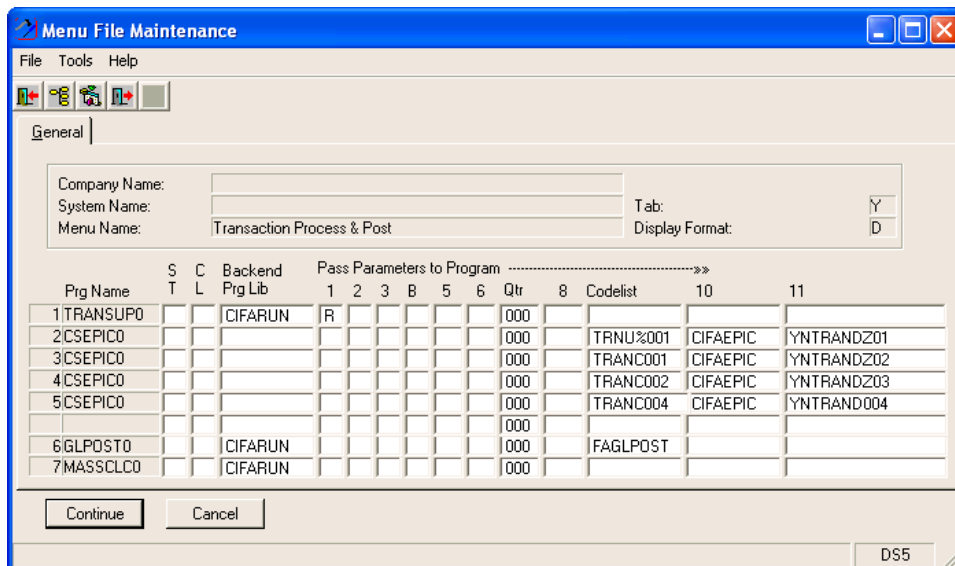
The changes to be made in this case are not security-related. The modifications can be accomplished in one of two ways:

- ◆ 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	Backend Prg Lib	Pass Parameters to Program								Codelist	10	11	
				1	2	3	B	5	6	Qtr	8				
1TRANSUP0			CIFARUN	R							000				
2CSEPIC0											000	TRNU%001	CIFAEPIC	YNTRANDZ01	
3CSEPIC0											000	TRANC001	CIFAEPIC	YNTRANDZ02	
4CSEPIC0											000	TRANC002	CIFAEPIC	YNTRANDZ03	
5CSEPIC0											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.

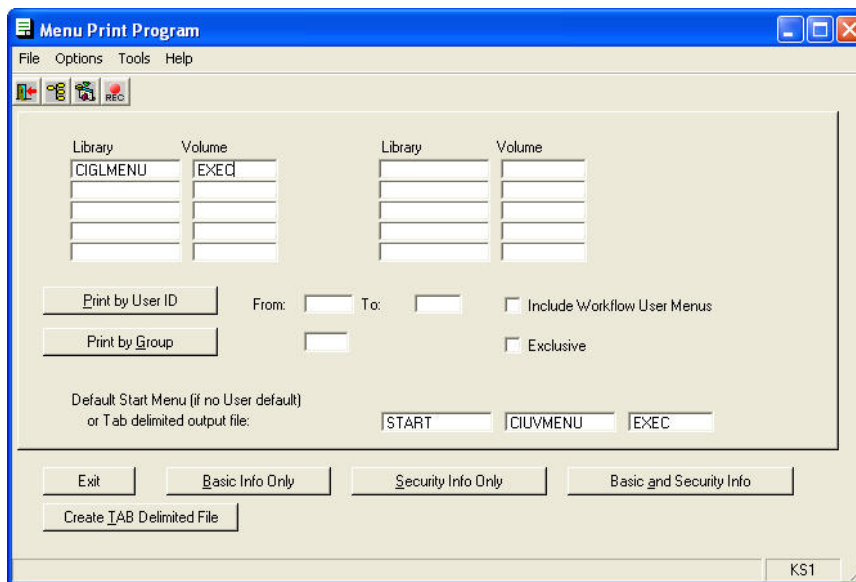
# 1.5 Menu File Print

## *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
<b>Print by User ID</b>	Print the menus to which each user in a particular range has access.
<b>Print by Group</b>	Print the menus to which a group has access.
<b>Exit</b>	Return to the Universal Utilities menu.
<b>Basic Info Only</b>	Print basic menu information. This prints the menus and the Other Programs Information.
<b>Security Info Only</b>	Print security information. This prints the name of the function or menu and the security information associated with each.
<b>Basic and Security Info</b>	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 ARMASTER 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 CIUVMENU     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? EN N 0 your print Queue

```

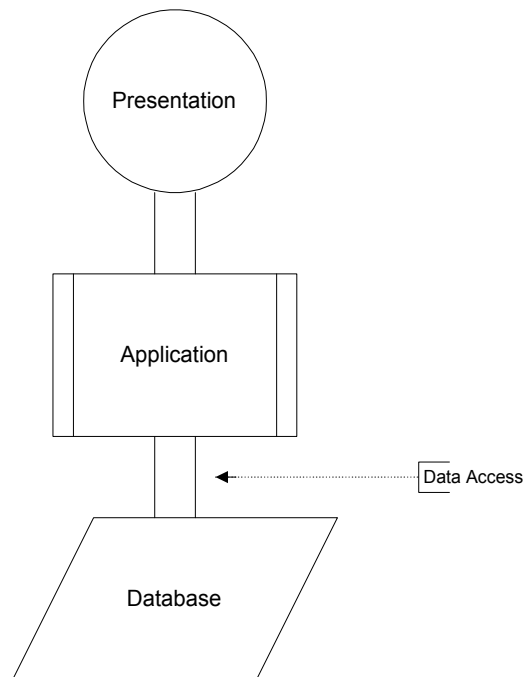


# 1.6 Database Map File Maintenance

## *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**

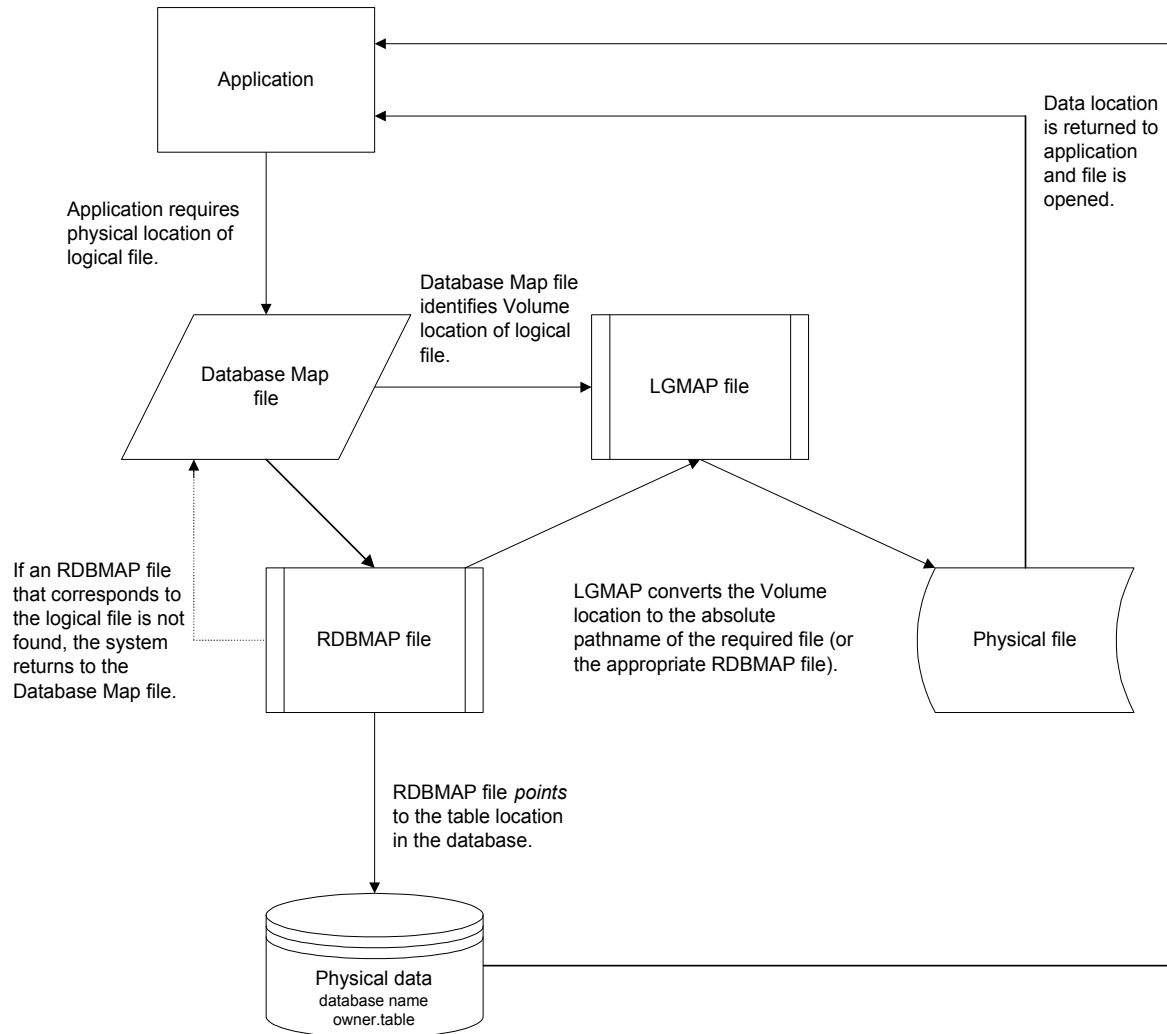


Computron applications locate data by reading the Database Map file in conjunction with the LGMAP file. An application program identifies required data by a logical name, which is comprised of a logical filename and a logical library. The Database Map file defines the physical location of logical files 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.

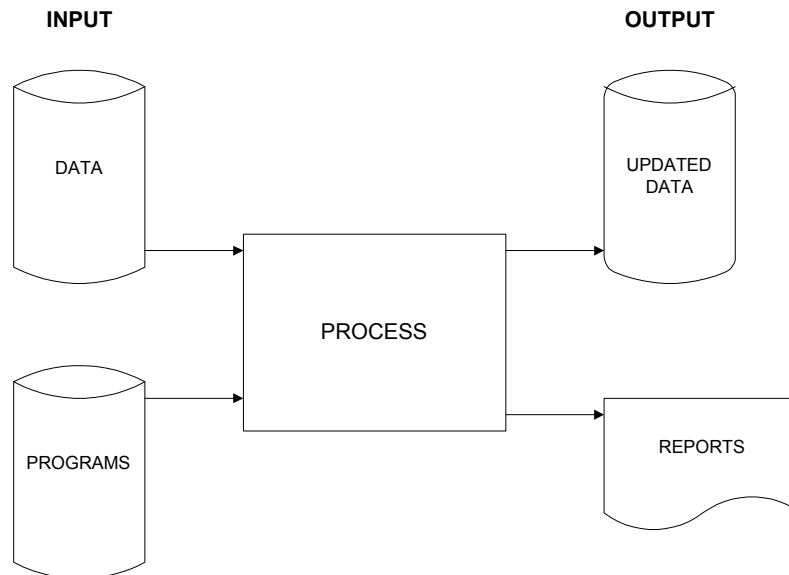
**Fig. 1.6-2**  
**Data Access Flowchart**



## 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 Windows 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 Windows 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 Windows 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 of Database Map Entries

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 /ctron file system, the LGMAP file entry for the volume name, DATA, appears as follows:

```
DATA /ctron/data
```

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

```
/ctron/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		Logical File	Description
	Library	Volume		
&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 Windows 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:

<b>&amp;UT</b>	The User Task number (3 positions).
<b>&amp;UT#</b>	The User Task number (4 positions).
<b>&amp;ID</b>	The User ID (3 positions).
<b>&amp;DATE</b>	The System data (6 positions or 5 positions if concatenated with &ID or &*UT).
<b>&amp;TIME</b>	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		Logical File	Actual File
	Library	Volume		
CIGL	&WORK	WRKVOL	EDETRANS	&UT#EDE

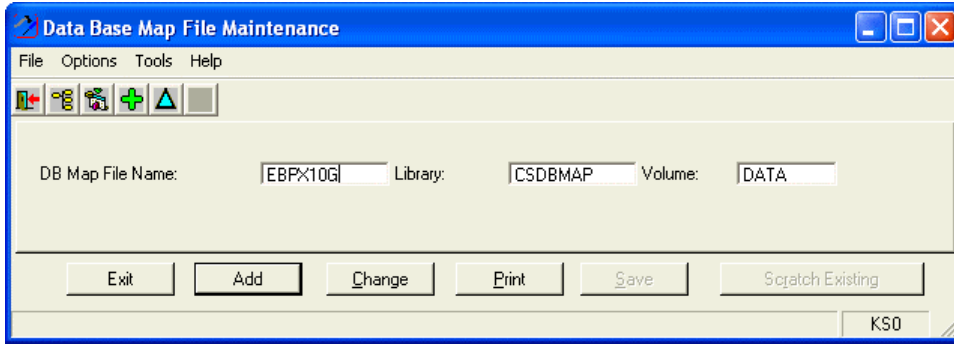
If the current User Task number is 3623, the actual filename created is 3623ede, which appears in the Native Location column.

### *Sample Actual File Entries*

Logical Library	Actual		Logical File	Actual File
	Library	Volume		
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.

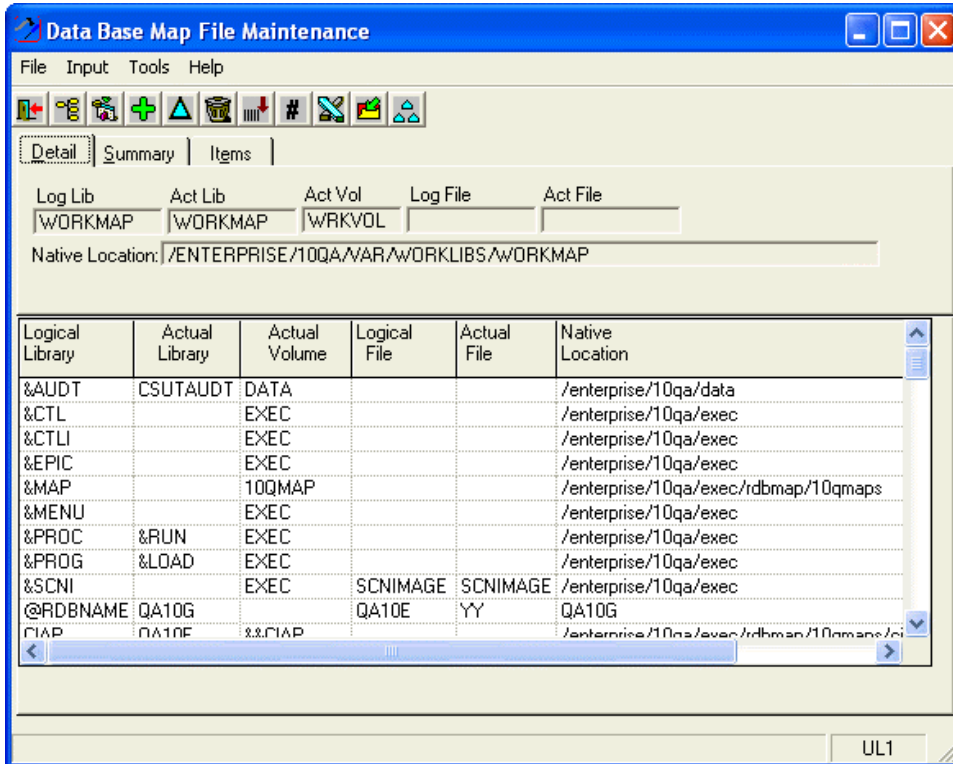
Table 1.6-1 Options for the Database Map File Maintenance Window (KS0)	
Button	Function
<b>Exit</b>	Exit the Database Map File Maintenance function.
<b>Add</b>	Add a database map. Proceed to the Database Map File Maintenance Window (DS1) where you can begin filling in the highlighted fields on the window.
<b>Change</b>	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.
<b>Print</b>	Print the Database Map File.
<b>Save</b>	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.

<b>Table 1.6-1 Options for the Database Map File Maintenance Window (KS0)</b>	
<b>Button</b>	<b>Function</b>
<b>Scratch Existing</b>	<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> <p style="text-align: center;">The Database Map File Has Not Been Saved</p> <p>When you click the Save button on the Database Map File Maintenance Window (KS0), a message displays in the status bar:</p> <p style="text-align: center;">File already exists press PF17 (Scratch Existing) to scratch it.</p> <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).



Logical Library	Actual Library	Actual Volume	Logical File	Actual File	Native Location
&AUDT	CSUTAUDT	DATA			/enterprise/10qa/data
&CTL		EXEC			/enterprise/10qa/exec
&CTLI		EXEC			/enterprise/10qa/exec
&EPIC		EXEC			/enterprise/10qa/exec
&MAP		10QMAP			/enterprise/10qa/exec/rdbmap/10qmaps
&MENU		EXEC			/enterprise/10qa/exec
&PROC	&RUN	EXEC			/enterprise/10qa/exec
&PROG	&LOAD	EXEC			/enterprise/10qa/exec
&SCNI		EXEC	SCNIMAGE	SCNIMAGE	/enterprise/10qa/exec
@RDBNAME	QA10G		QA10E	YY	QA10G
CIAP	QA10E	&&CIAP			/enterprise/10qa/exec/rdbmap/10qmaps/ci

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.

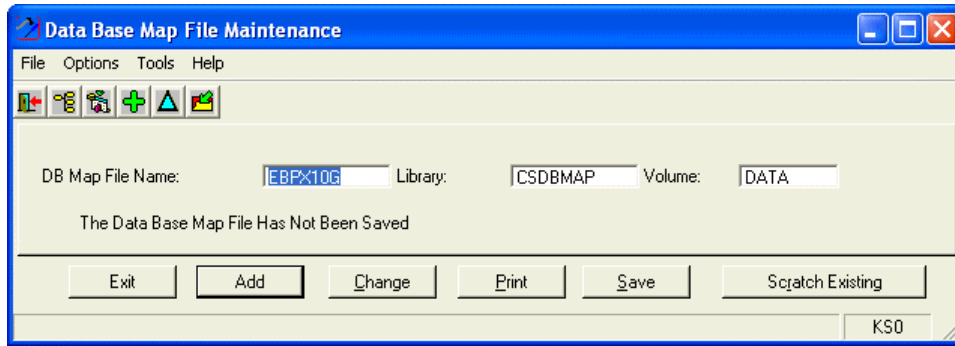
### Native Location

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

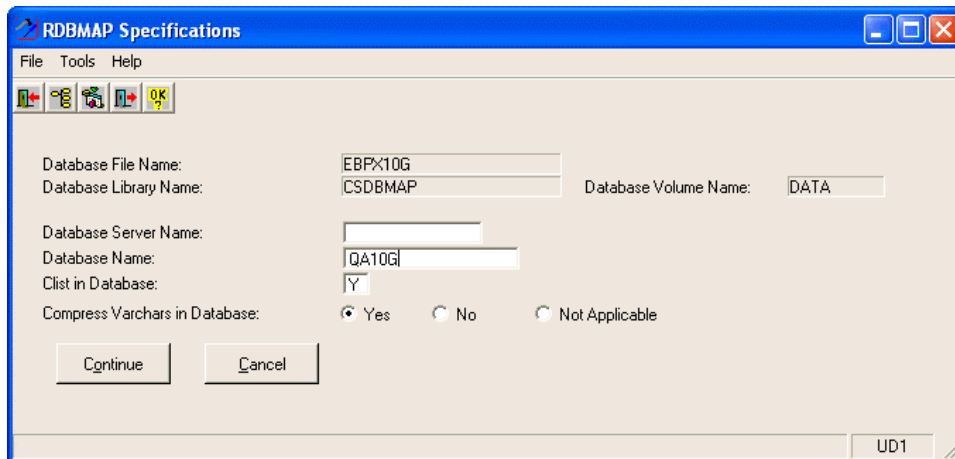
For RDBMS implementations, some additional database map parameters can be established on the RDBMAP Specifications Window (UD1). For instance, some implementations may need to update older versions of their database maps in order to access the database tables associated with code list (CLISTCTL) RDBMAPS. 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 KSO 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.

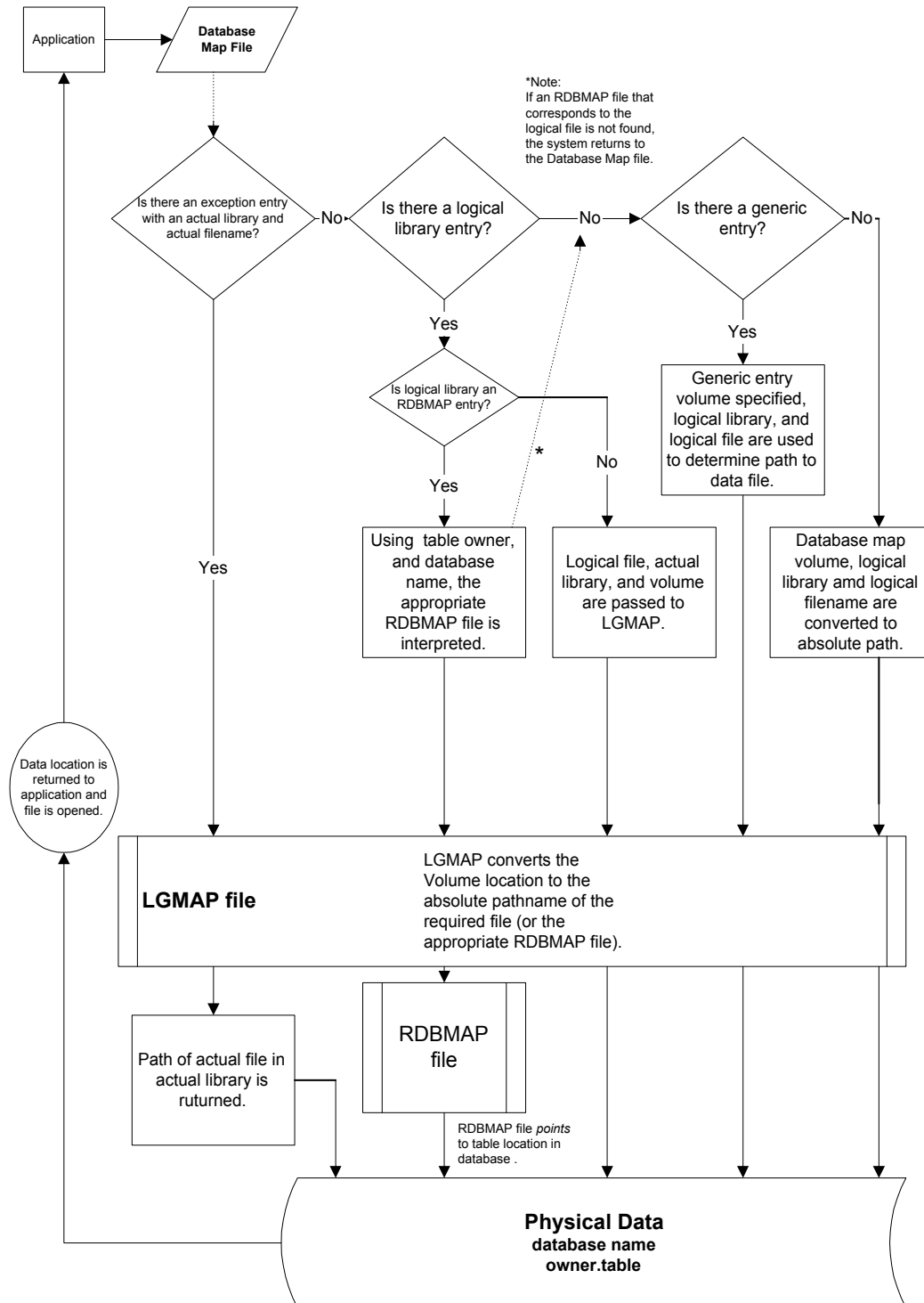
The program always searches the logical library encountered first in the database map for the required data. Thus, the program first interprets the RDBMAP file to determine the table location where the required data is stored. If found, the table information (table name, table owner, column, and row) 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

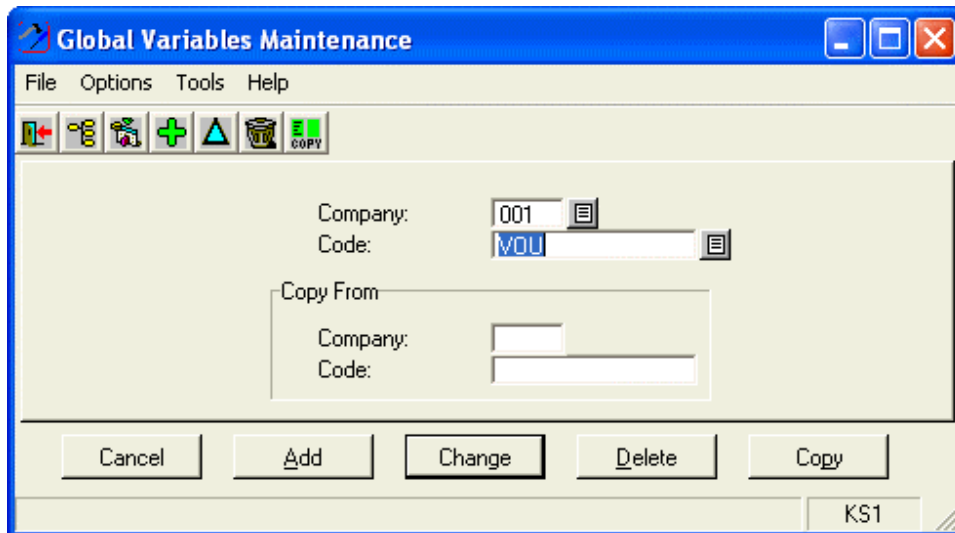
after the number 999 is used, the system assigns the number 1 to the next record.

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.

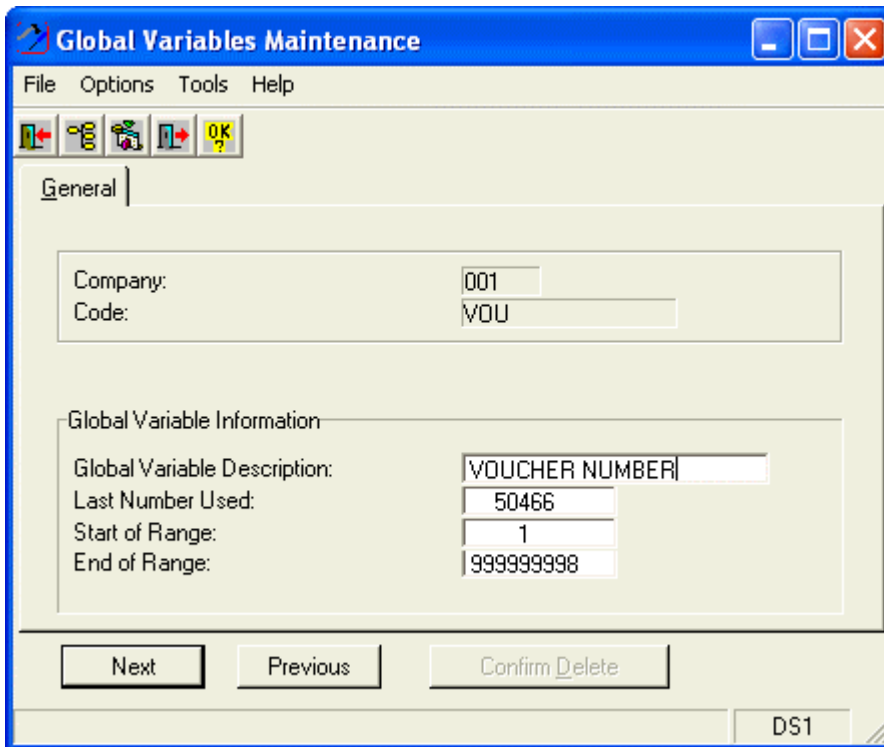
### Copy From Company Code:

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

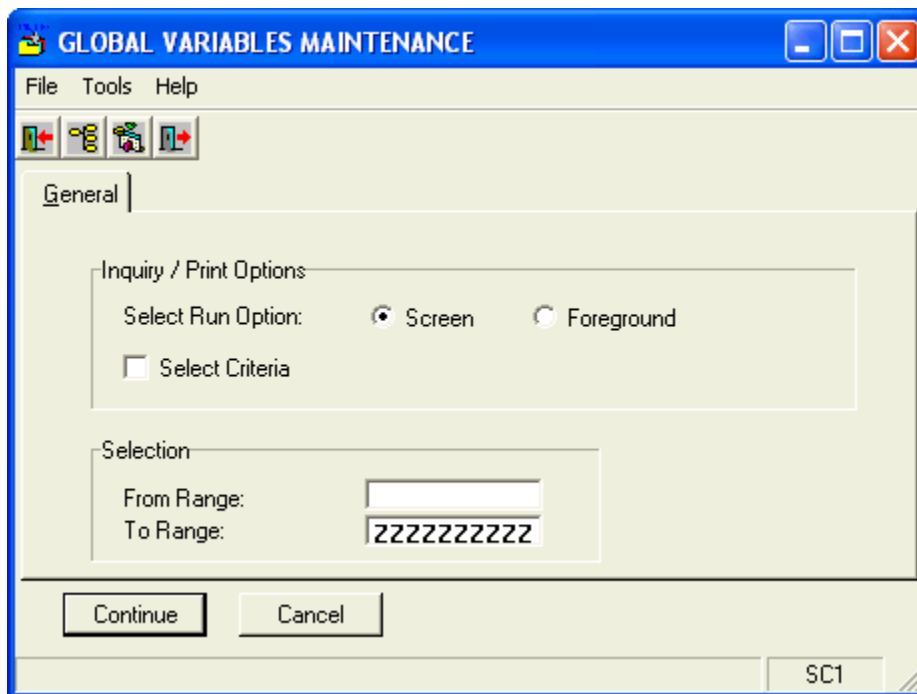
Maximum system-generated number assigned to a transaction using this global variable. If this number is reached, the system assigns the Start of Range number to the next transaction. In the example shown on the above window, once you enter the transaction number 999,999,998, the system assigns the next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

# 1.8 Global Variable Inquiry/List

## *Introduction to Global Variable Inquiry/List*

This function displays 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.

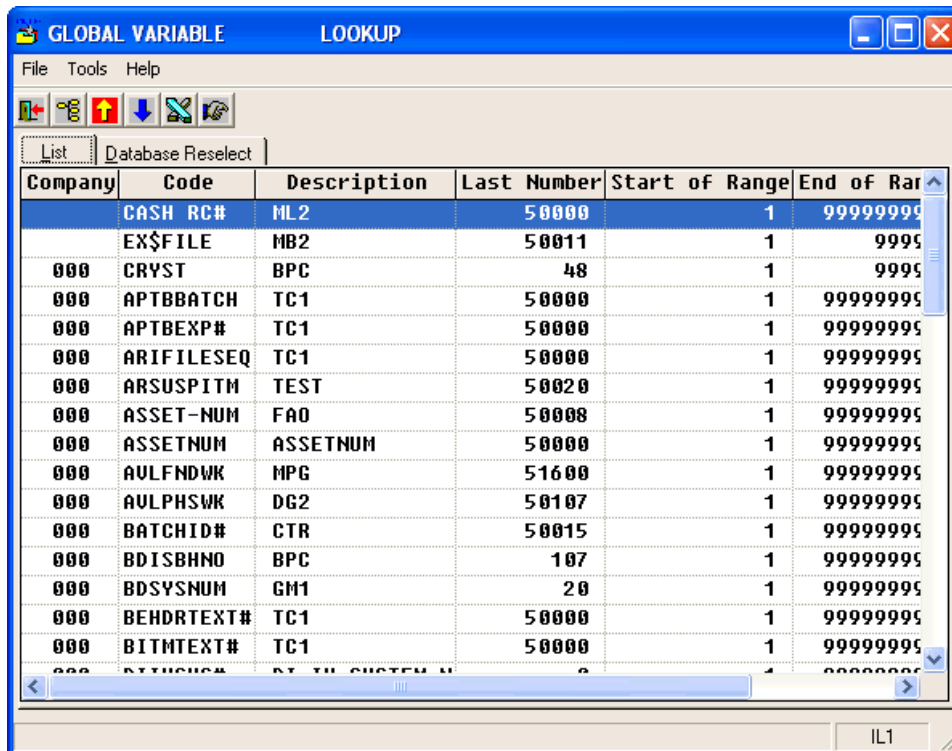
### To Range

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

## Global Variable Report Description

### Global Variable Screen Option

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



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	APTBATCH	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	BDSYSNUM	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:

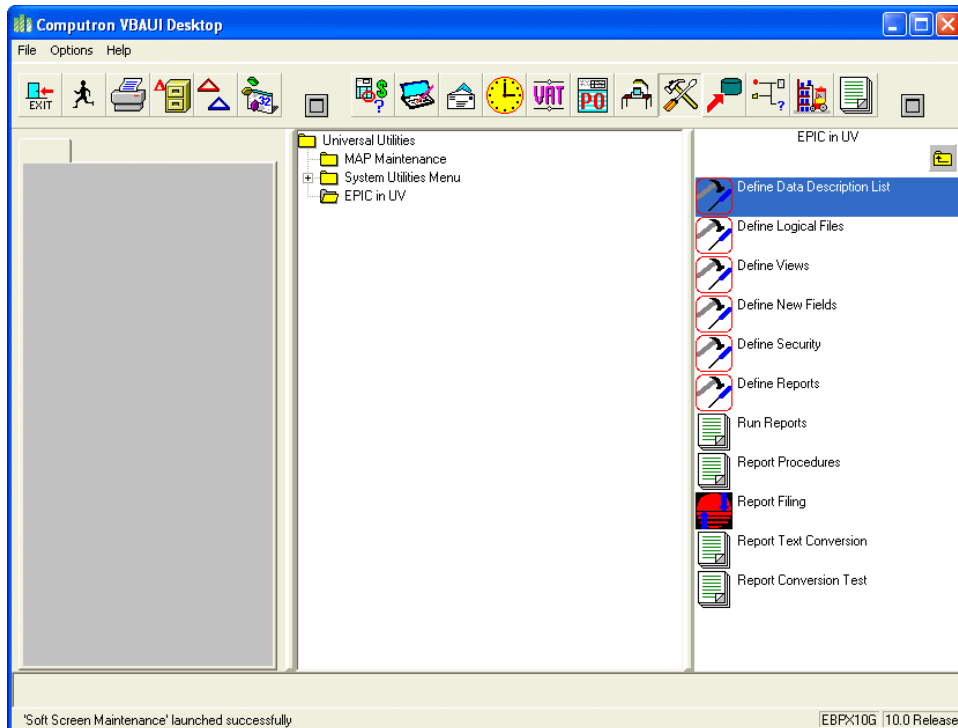
- ◆ highlight it and click the Choose button;
- ◆ double-click the desired code.



# 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.

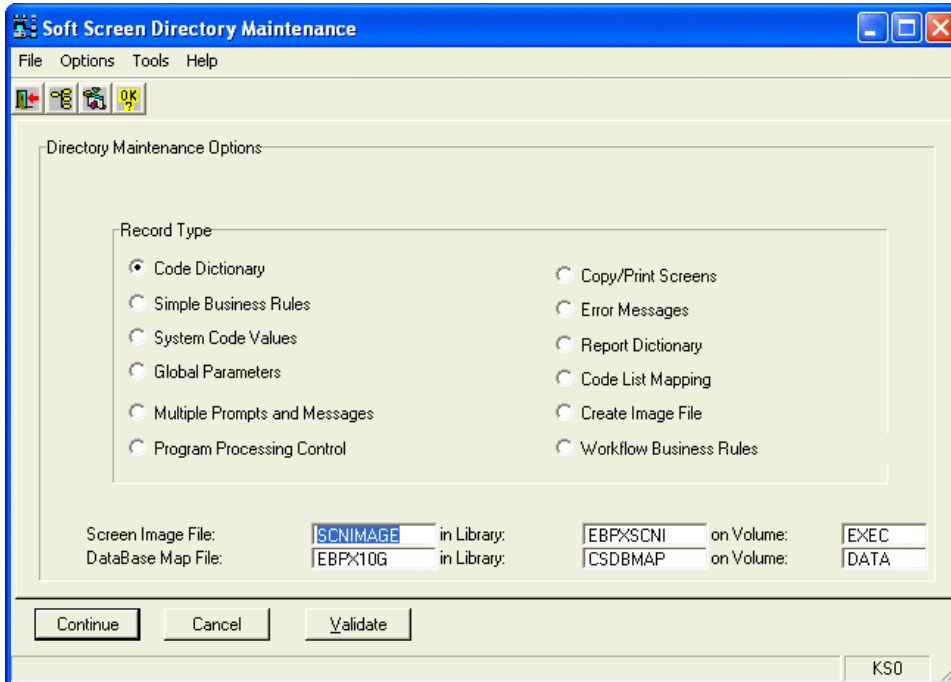


**EPIC Report & Query Generator Menu**

# 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 (KS0)**

# 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*

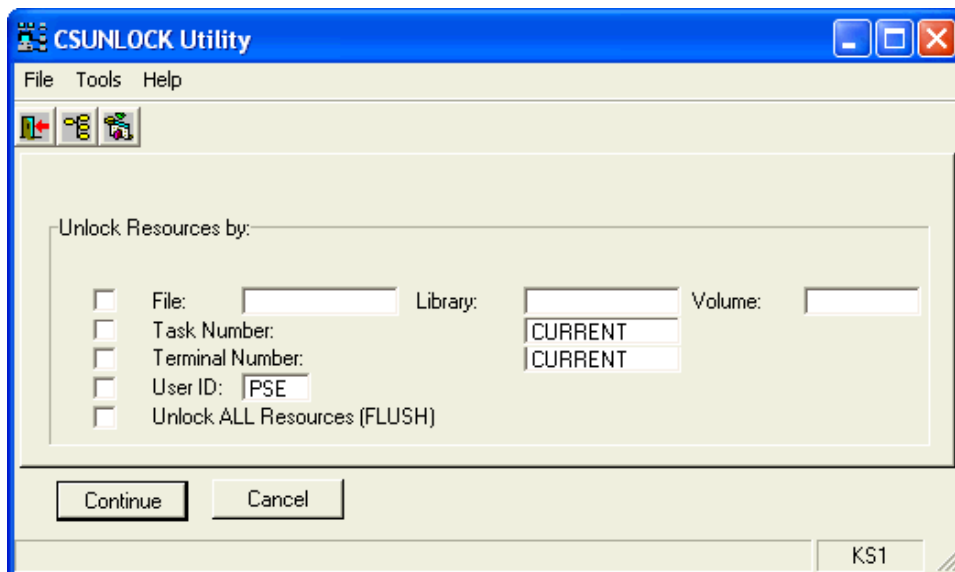
When a program needs to update tables, resources are held. This is necessary so that two programs competing for the same resource don't enter into an infinite lock and halt processing. In normal processing, resources are released when the function is complete. In case the system malfunctions, held resources 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)**

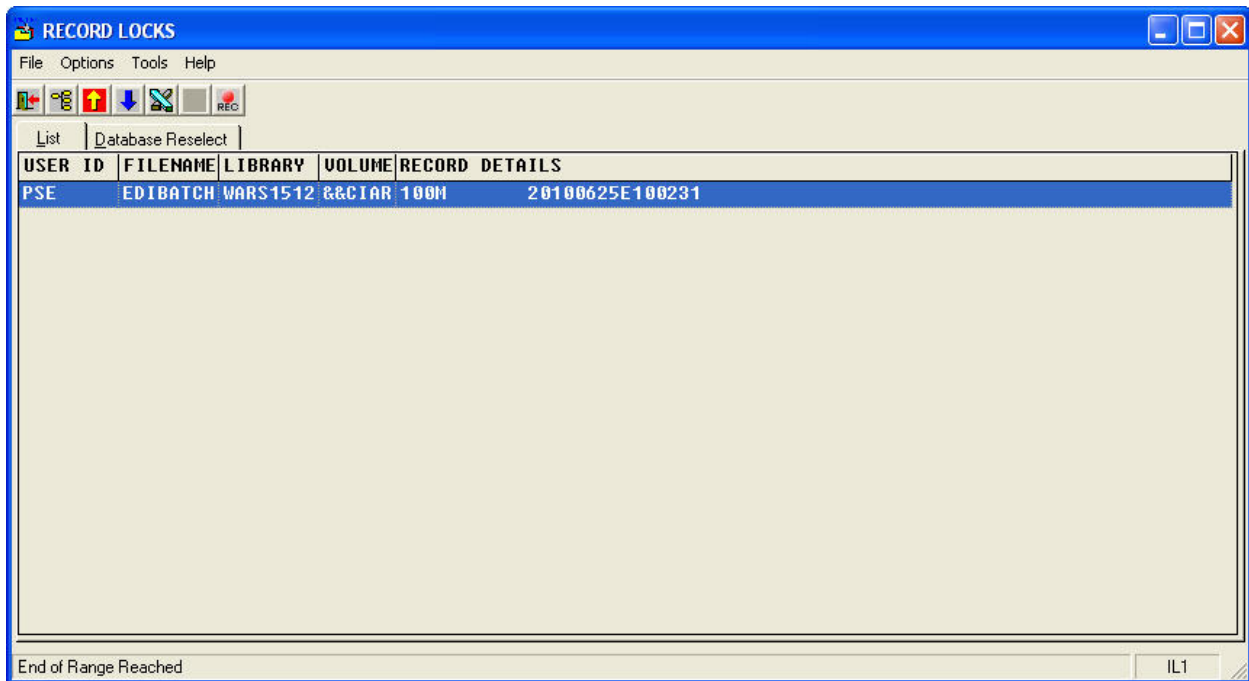
# 1.13 Lockout Inquiry

## *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)**



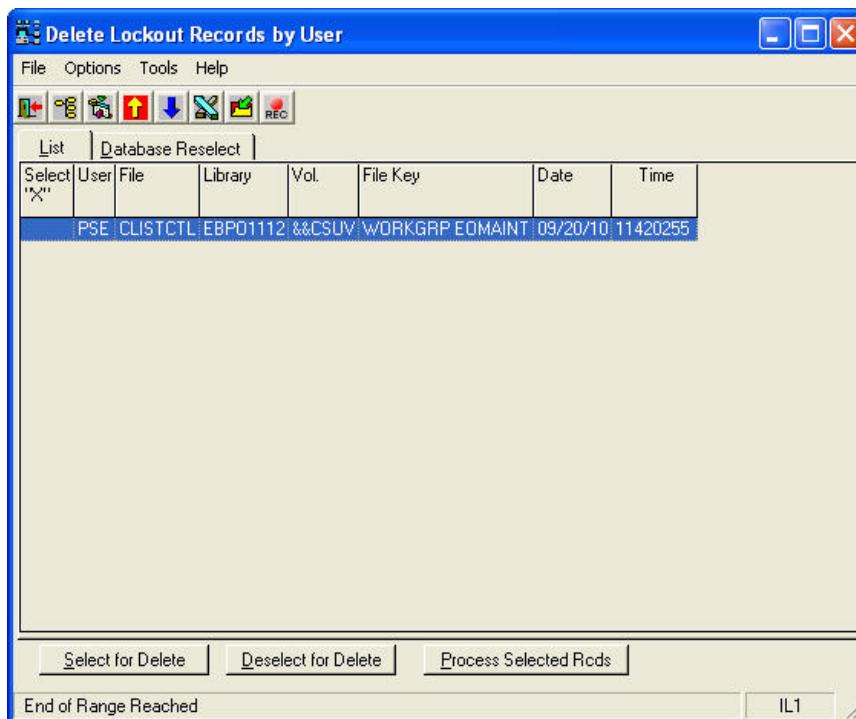
# 1.14 Lockout Delete by User

## *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
<b>Select for Delete</b>	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.
<b>Deselect for Delete</b>	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.
<b>Process Selected Records</b>	This unlocks the records that have been selected and returns you to the menu.

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

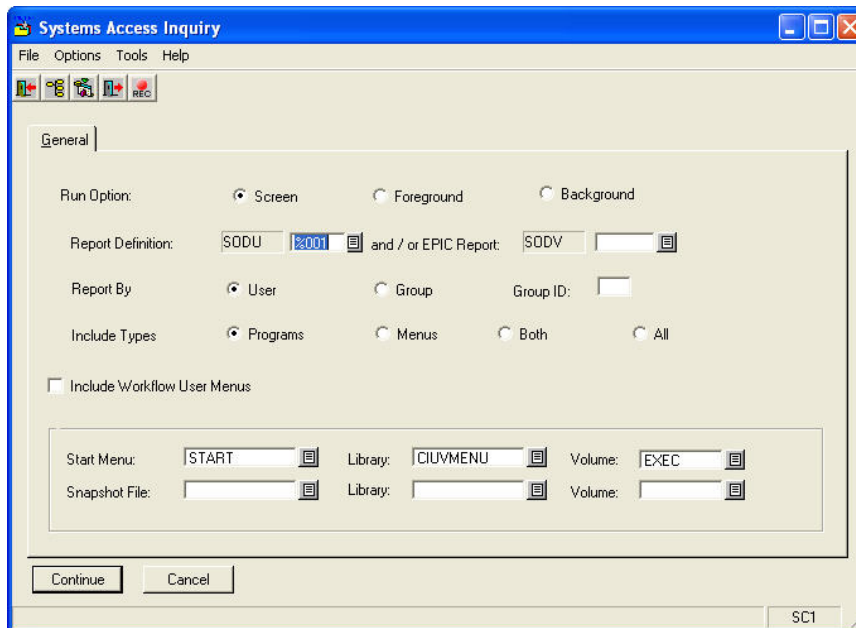
### *How the Information for the Report is Determined*

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.

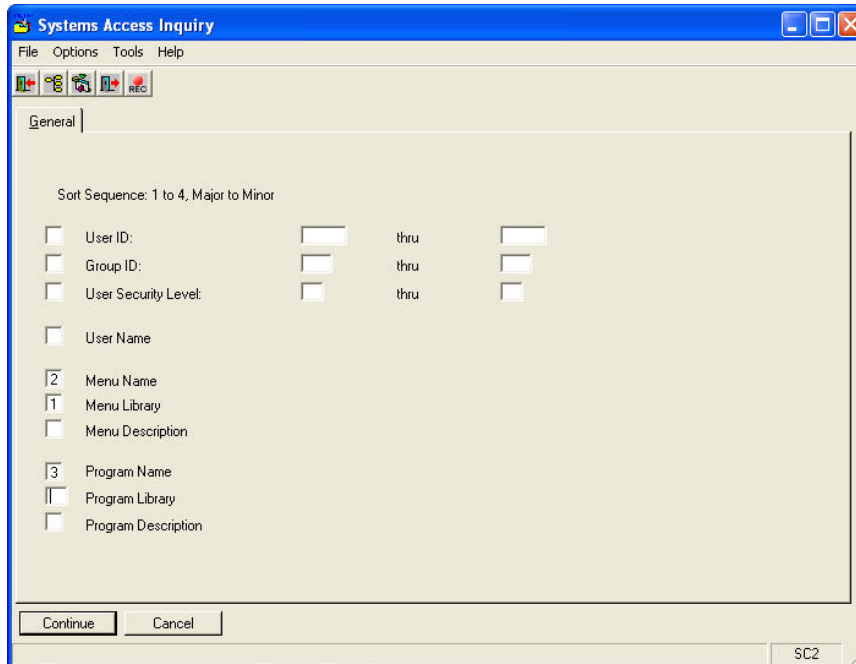
☞ *Note:* Due to the extremely large number of menu options in the full Computron system, use this field to limit the extent of the report.

### 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

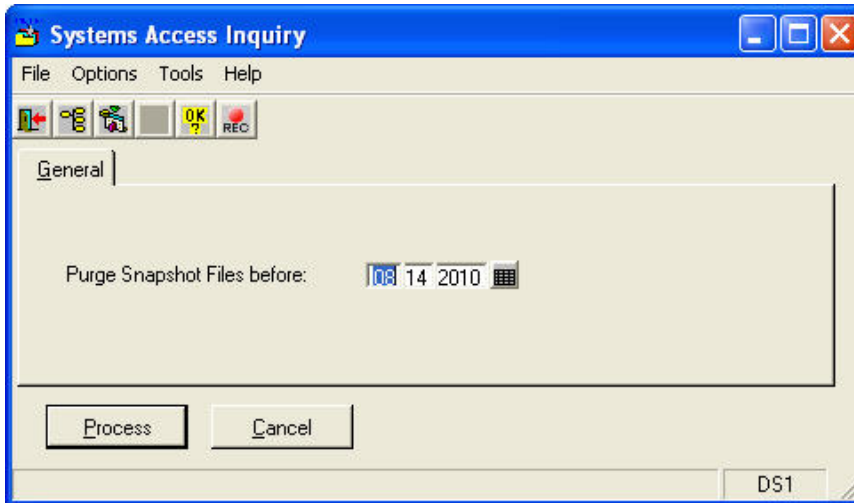
\*\*\*\*\* Mon Sep 13, 2010 2:45 PM \*  
\*  
\* Systems Access Report \*  
\*  
\*\*\*\*\* Page 3 \*

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	TAPF1042	CIAPLOAD	GENEX-Out Vendor 1042 Dat	P	0	APEXTRNL	CIAPMENU GENEX -- External Interface
PSE	Pat Eckart-McGovern	9	TAPF1099	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*

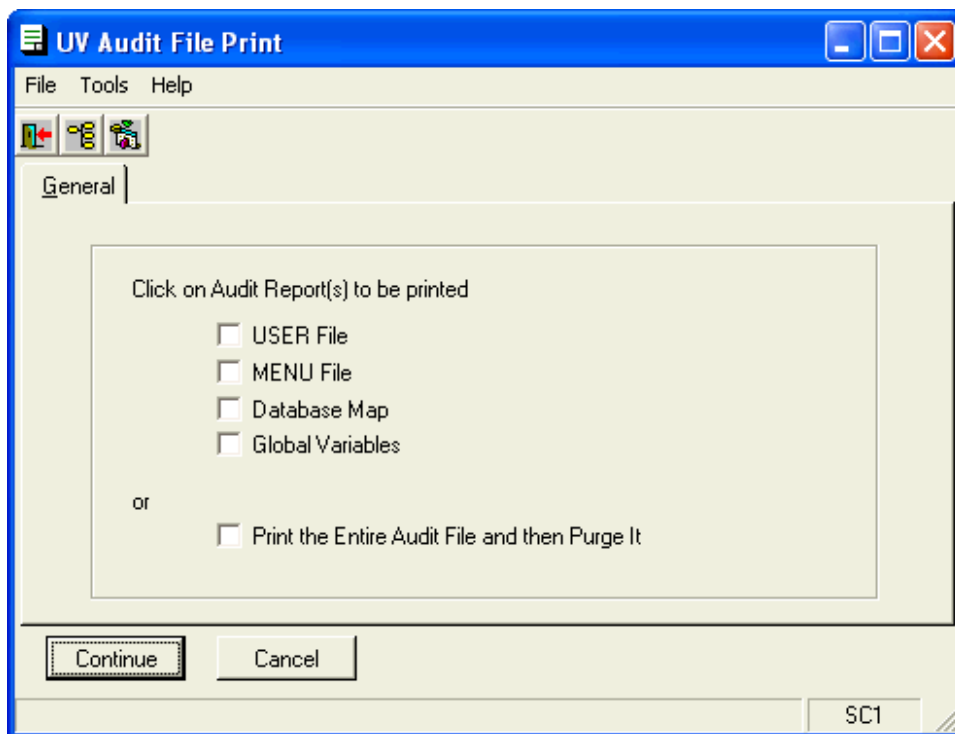
This function gives you the option to print a report on the following files:

- ◆ 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



### 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	WPAR10G	DATA		
CIWF	BPA10	&&CIWF	CONTACT	CLISTCTL
CIWF	WPAR10G	DATA	WFTASK	CLISTCTL
CIWFCTL	WPARCTL	EXEC	SYSVAR	SYSVAR
CIWFMENU	WPARMENU	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	BPA10	&&WPAD		
WPAD	BPA10	&&CIWF	CONTACT	CLISTCTL
WPAR	QA10D	&&WPAR		
WPAR	WPAR10G	DATA		
WPAR	QA10D	&&WPAR	CLISTCST	CLISTCST
WPAR	QA10D	&&CIWF	CLISTCTL	CLISTCTL
WPAR	BPA10	&&CIWF	CONTACT	CLISTCTL
WPAR	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 Windows system, a relational database or the Computron applications, users must supply information that identifies them to the system.

#### Windows Authentication

Access to the Computron applications begins with access to the Windows system where the Computron software resides. When logging onto the operating system, the user must supply a Windows user name and password. The Windows Security subsystem then verifies the user account in the Windows user database, as follows:

- ◆ The user name must specify a current user account.
- ◆ The password must be correct for that account and cannot be expired.
- ◆ The account must not be flagged as *User Must Change Password at Next Logon*.
- ◆ The account must not be disabled.
- ◆ Other Windows restrictions can be applied, such as setting certain hours when the user can log on. (Refer to your Windows Server documentation for details.)

The Windows user account also establishes system-wide security privileges for the user, such as assignment to a particular user group and/or Windows domain. 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).

#### Computron Logon Authentication (via the CSServe Service)

Once a user accesses the Windows Server, the user must be validated within the Computron system. The main Computron service, *CSServe*, is responsible for processing connection requests. An instance of this service is created for each configuration created with the Computron Control Panel during the setup program. Each configuration, when started, runs an instance of *CSServe*. Each instance of *CSServe* listens for connection requests on one or two TCP/IP ports (the main port and, optionally, the Telnet port).

*CSServe* processes Computron AUI and character cell connection requests. With regard to security, *CSServe* handles each type of connection request identically. Currently, Computron software requires lowercase Windows user IDs and passwords. New installations have an eight-character (8) limit for user IDs; while existing installations require a five-character (5) limit to accommodate print queue IDs.

### *For AUI Clients*

The user enters their user name (ID) and password. The password is optionally encrypted. The AUI client software requests a connection to CSServe on the Main Port TCP/IP port number specified during the Computron installation procedure. CSServe receives the connection request, which contains the user name, password, and other information. If necessary, CSServe decrypts the password.

### *For Character Cell*

For clients running character-based sessions of Computron software, the Telnet client software connects to CSServe on the character cell TCP/IP port number entered during the Computron Software installation procedure. CSServe recognizes that this is a character cell connection, so it prompts for a user name (ID) and password.

☞ *Note:* For detailed instructions on configuring and running character cell under Windows, refer to Appendix A, “Configuring Character Cell for Windows.”

### *For All Client Types*

Once CSServe has the user name (ID) and password, it then authenticates the user. It asks the Windows operating system to verify this user’s credentials. Windows then checks the password. Windows ensures that this user is allowed to log on (if account is not disabled, etc., as described earlier). If the logon information is correct, then CSServe launches an application process on behalf of that user, with the security privileges of the user (i.e., it can only do what that Computron user is authorized to do). After the application process is created, it inherits the socket created by CSServe, and then the service returns to the task of listening for connection requests.

### *Ntstart Sessions*

The ntstart utility allows administrators to access Computron system tools (e.g., vutil) outside of the normal operating environment. When ntstart.exe is launched, it opens a command-line Windows console session that checks the system configuration settings, which in turn, establishes environment variables, creates a work library, and invokes Computron’s security subsystem.

☞ *Note:* For detailed information about the Ntstart utility, refer to Section 4.2, “Computron’s Ntstart Utility.”

## Validate and Impersonate Settings

This section describes the function of Computron’s user authentication configuration settings: *VALIDATE* and *IMPERSONATE*.

- ◆ *VALIDATE* tells CSServe whether to authenticate the user.
- ◆ *IMPERSONATE* tells CSServe whether to run the process on behalf of the authenticated user.

These settings are maintained via the Computron Control Panel on the Windows console and, under normal operating conditions, they should both be set to TRUE.

VALIDATE = TRUE  
IMPERSONATE = TRUE

This setting tells CSServe to authenticate the user (check ID and password) and to run processes on behalf of that user.

***Why Use This Setting?***

This setting provides normal security. Users must be valid Windows users and, as in all cases, they must also be valid Computron users. Computron processes run on behalf of the user. The Windows authority of the particular user running Computron applications limits file and application access.

VALIDATE = TRUE  
IMPERSONATE = FALSE

This setting tells CSServe to authenticate this user (check his ID and password), but do not run processes on behalf of that user. This means that the user must supply a valid Windows user name (ID) and password. However, when CSServe creates the process to run the application, it does not create the process on behalf of that user. Instead, the process runs in the same security context as CSServe. This is usually the Local System account, which is a powerful administrative account.

***Why Use This Setting?***

Typically, you should not use this setting. However, on a nonproduction system, where perhaps a security scheme has not been fully devised and implemented, this setting will allow users to access Computron software without running into security problems. All valid Windows users who have also been defined as valid Computron users have full access to Computron software. You can place security restrictions on data and program files so that the general population does not have access. Because the Computron users are running in the Local System account, they still have full access to the system, but only through Computron software. While this may sound like an acceptable security scheme, there are other better ways of accomplishing this type of security (like specifying a powerful Windows group in the CtronGroups configuration setting).

VALIDATE = FALSE  
IMPERSONATE = *anything*

By setting VALIDATE to FALSE (and therefore implying IMPERSONATE = FALSE) you are telling CSServe not to authenticate the user. This literally means that users can specify any user name (ID) and password. The user name does not have to be a valid Windows user, nor does Computron software ask the Windows operating system to check the user name and password. Logically, it follows that processes cannot be run on behalf of the user (because the user may or may not be a valid Windows user and has not been checked). Processes then run in the same security context as CSServe, usually the Local System account. As described earlier, this is a very powerful account that has the ability to access just about anything on the machine.

***Why Use This Setting?***

Once again, you should not use this setting under normal operating conditions. This setting would only make sense in a purely test environment where security isn't an issue. Computron security is always in effect; therefore, you should always create Computron users. (The user name specified upon logon must be a valid Computron user, but doesn't need to be a valid Windows user.)



## 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 Windows 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 Windows 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 Windows system security and the RDBMS security establish user access privileges at the table level (RDBMS) and file level (Windows), 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 Windows 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 Windows,” at the end of this section.

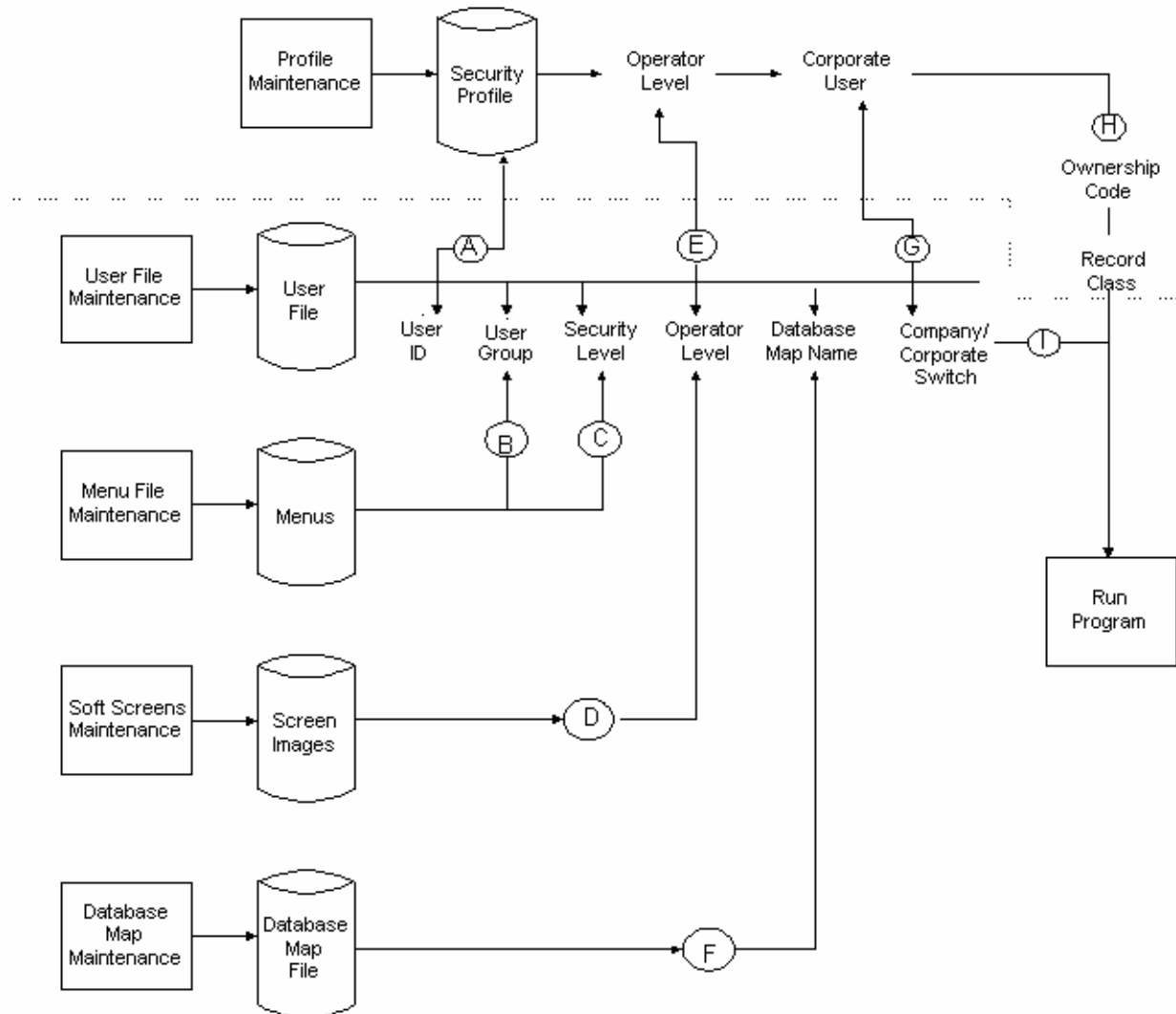
## *The Security Mechanism*

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.

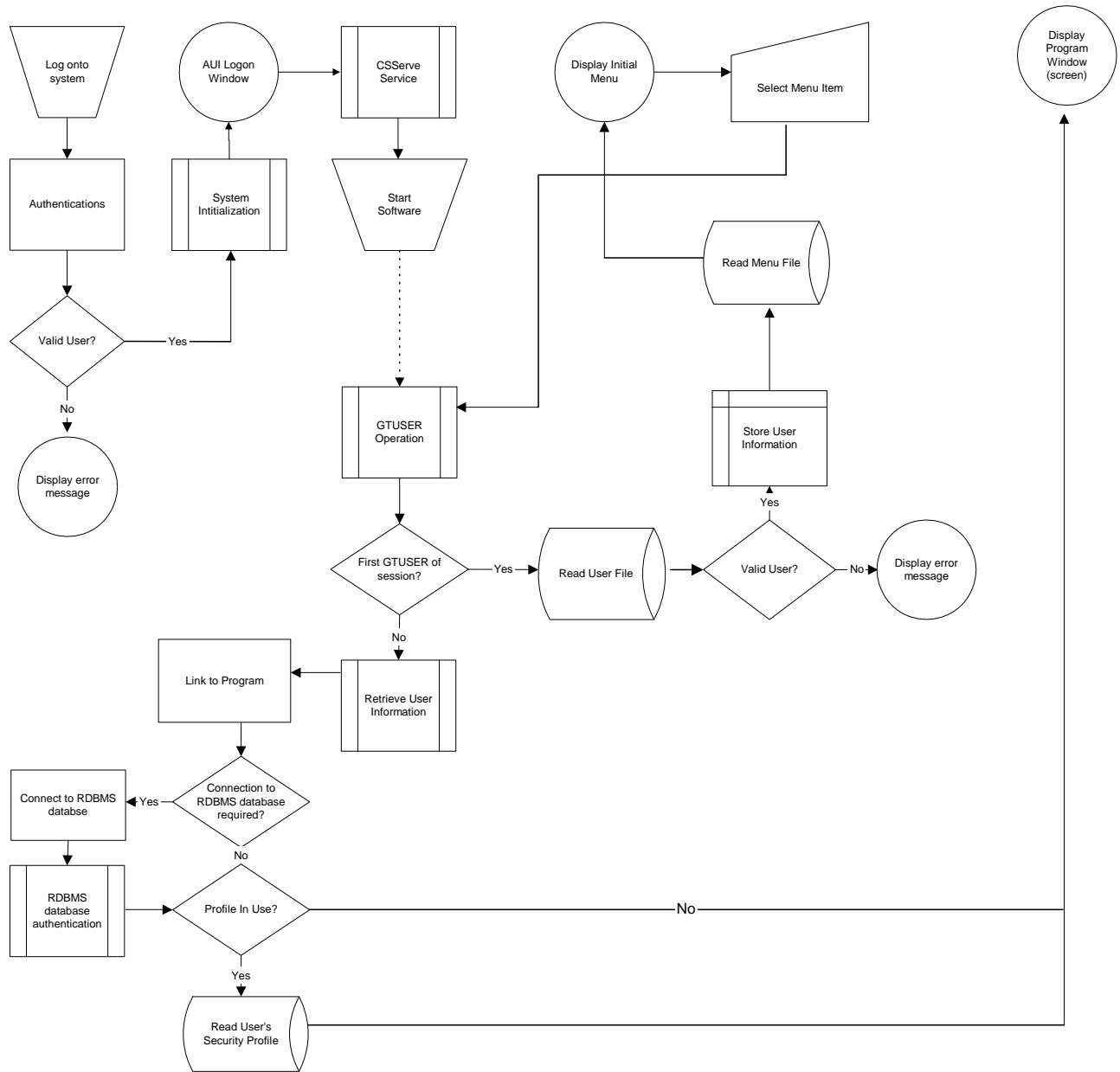
The impact of the various components of the security subsystem on the user's actions that begin with logging onto the system are illustrated in the Security Subsystem Processing flowchart later in this section. This flow chart denotes each step performed by the user and each step consequently performed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

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

## User Logs onto the Windows System

1. The Windows security validates the user based on the Windows current user account.
  - If the user does not enter a valid password, Windows 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 Windows 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. (This could be either WindowsStart or a non-Computron 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 service 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 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. Before actually running an application, the process verifies that the user is a valid Computron user by querying the Computron User file (USERFIL3), checking for a record that links the Windows user account to an 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 all 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 Windows

The table below represents the minimum Windows 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`. Keep in mind, however, that every installation is unique and therefore, may require specific permissions for other users.

Directories/Files	Permission		Access Options
	Dir.	Subdirs.	
\...\	RWX	RWX	Special Access (add W)
\...\data	RWXD	RWXD	Change
\...\exec\xxxxscni	RWXD	RWXD	Change
\...\exec\xxxxmenu	RWXD	RWXD	Change
\...\exec\xxxxepic	RWXD	RWXD	Change
\...\exec\worklibs	RWXD	RWXD	Change
\...\exec\_ctron_\userfil3 (file only)	RWXD		Change
\...\exec\_ctron_	RX	RX	Read
\...\exec\xxxxload	RX		Read
\...\exec\xxxxrun	RX		Read
msnt	RX		Read
temp	All		Read
tmp	All		Full Control
temp2 (if exists)	All		Full Control

In addition to the minimum ctron group permissions described above, Computron suggests the following administrative permissions:

- ◆ The local Windows administrator must have Full Control for the entire Computron Software system – even if Computron is installed on a domain server
- ◆ The Computron administrator needs Full Control to the `\...\exec\_ctron_` directory, in order to stop and start services without rebooting the server.

## Print Queue Security

The first time a Computron user generates a report (e.g., Jones), the system automatically creates a personal print queue Spool that corresponds to the user's ID (e.g., `\...\var\spool\jonesprt`). Keep in mind however, that in the Windows Server environment these Spool subdirectories are not secure by default. This means that all users would have full access to any report in any print queue directory.

To ensure print queue integrity for individual users, you should set the Windows permission on the Spool directory for the ctron group as follows:

Directory	Permission		Access Options
	Dir.	Subdirs.	
<code>\...\var\spool</code>	RX	RX	Read

In order to have someone who can administer all of the print queue directories, you should also add the Local Window Administrator, with Full Control to the Spool directory.

Next establish print queue subdirectories for each user account and then set directory permissions for each user as follows:

Special Files	Permission		Ownership
	Dir.	Subdirs.	
<code>\...\var\spool\xxxxprt</code>	All		Full Control

Now when users are browsing for reports in FileView, they only see the print queue directory that corresponds to their user ID

☞ *Note:* For step-by-step instructions on setting permissions for Computron print queues, refer to Section 1.1 “Overview: Setting Up Users.”

## 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.

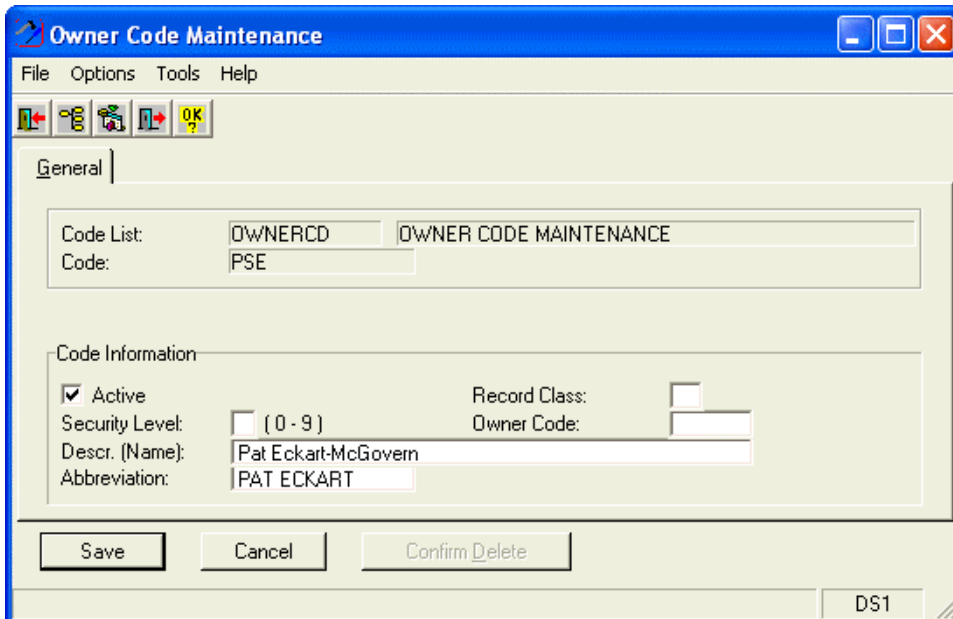
*Note:* On entry windows, the Ownership Code field is always Q-Markable.

### *Owner Code Maintenance Window (KS1)*

The system prompts for entry of an ownership code (Q-<sup>Mark</sup>). 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)*



The screenshot shows the 'Owner Code Maintenance' window with the following details:

- Title Bar:** Owner Code Maintenance
- Menu Bar:** File Options Tools Help
- Toolbar:** Includes icons for Add, Change, Delete, Copy, and a Q-Mark icon.
- General Section:**
  - Code List: OWNERCD | OWNER CODE MAINTENANCE
  - Code: PSE
- Code Information Section:**
  - Active
  - Security Level:  (0 - 9)
  - Descr. (Name): Pat Eckart-McGovern
  - Abbreviation: PAT ECKART
  - Record Class: [ ]
  - Owner Code: [ ]
- Buttons:** Save, Cancel, Confirm Delete
- Status Bar:** DS1

Ownership Code Maintenance Window (DS1)

**Active**

Enter the status of the ownership code. This field appears on Q-<sup>Mark</sup> 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-<sup>Mark</sup> windows.

**Abbreviation**

This is a ten position alphanumeric description of the ownership code. This field appears on Q-<sup>Mark</sup> windows.

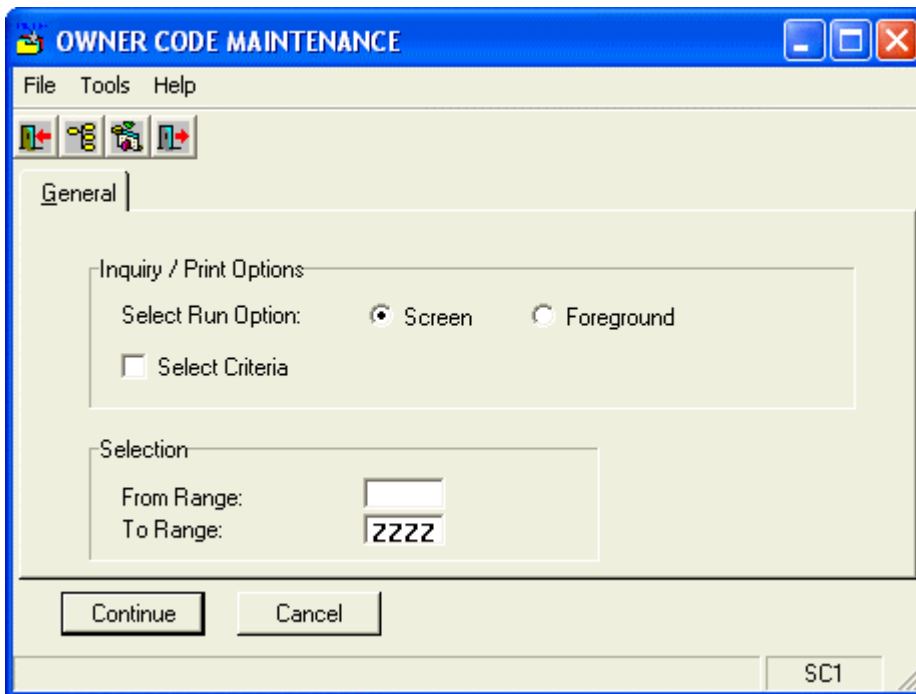
## 2.3 Ownership Code Inquiry / List

### *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.

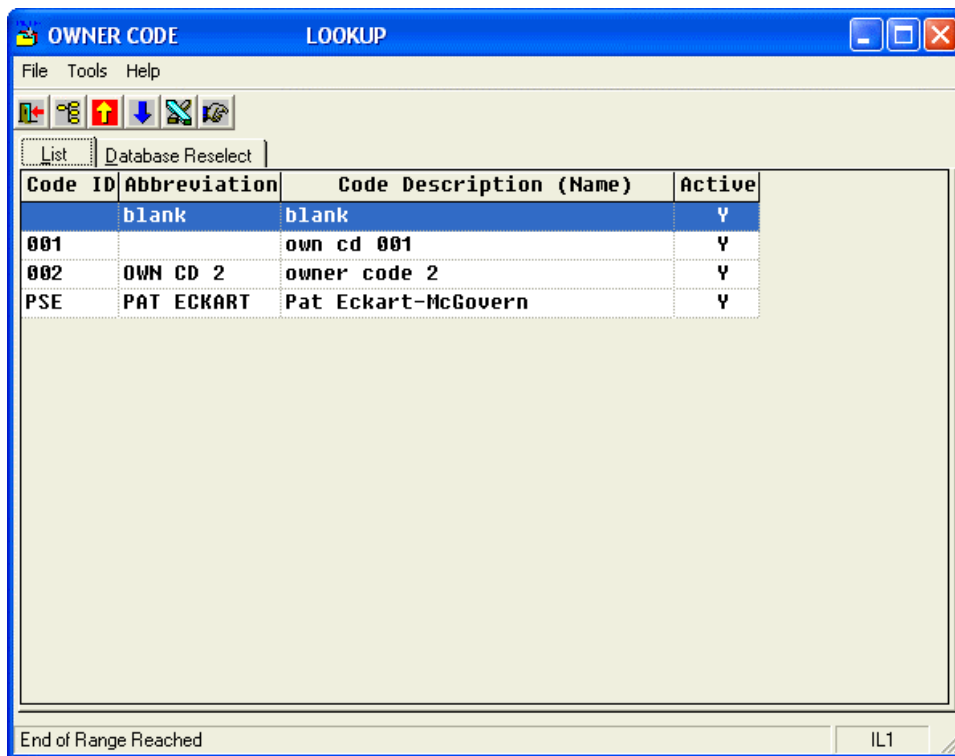
### To Range

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

## Ownership Code Report Description

### Owner Code Screen Option

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

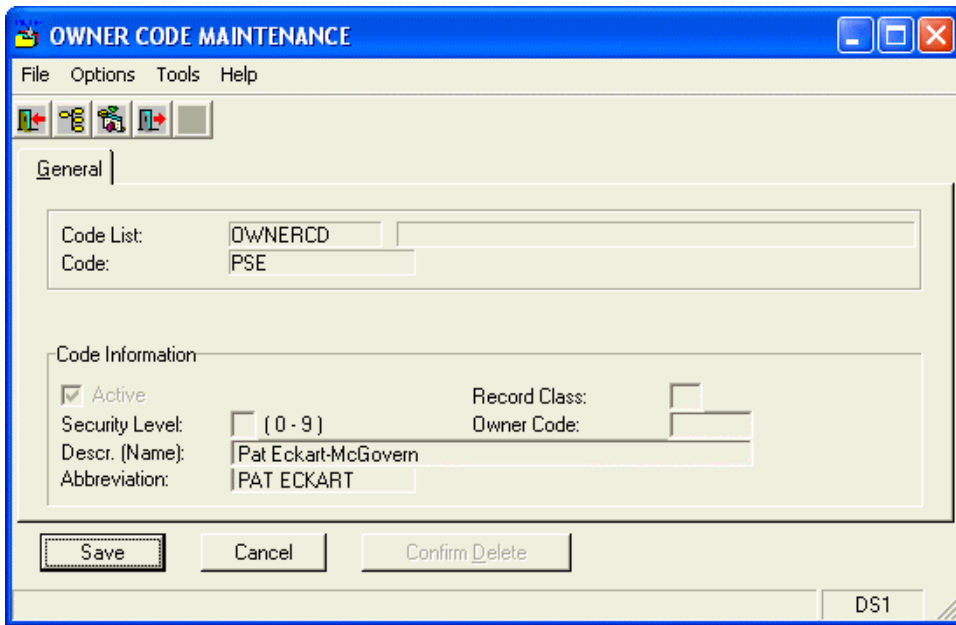
At the bottom of the window, there is a status bar with the text "End of Range Reached" and a small box containing "IL1".

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**

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)

## Ownership Code 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:

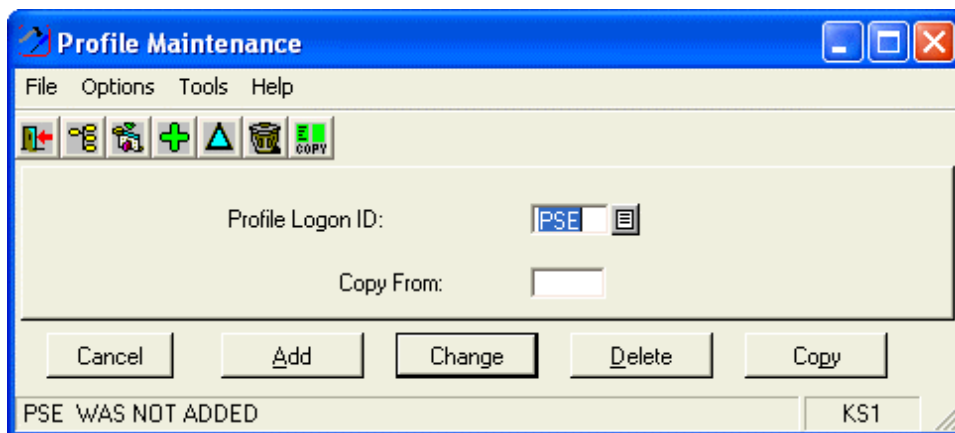
Allow/Disallow	From — To
D	AAAA — AAAA

☞ *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)

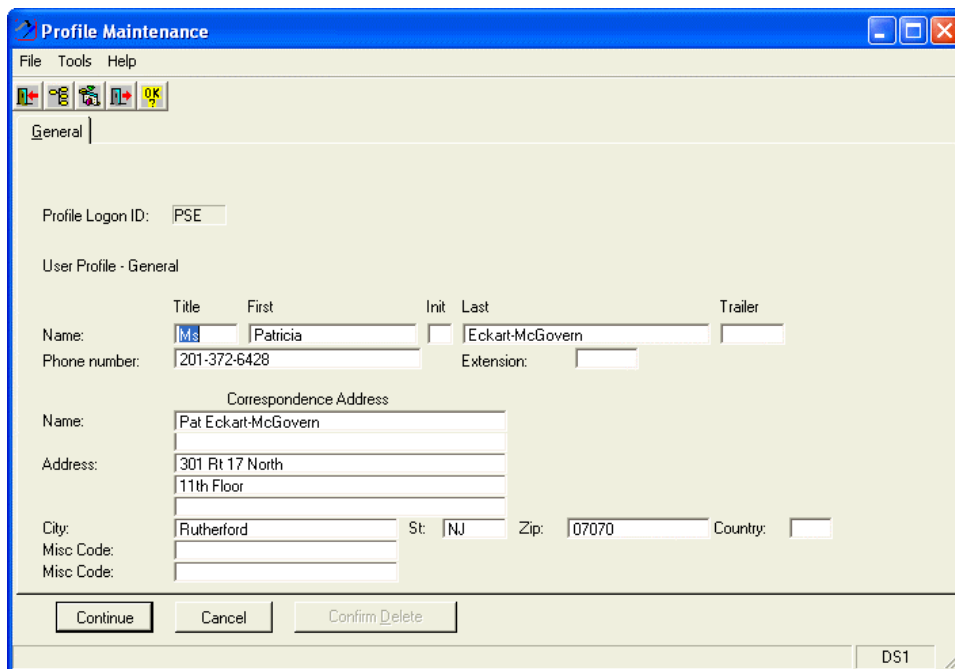
### Profile Logon ID (Q-Mark)

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 been 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.

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



## 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-<sup>Mark</sup>)**

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.

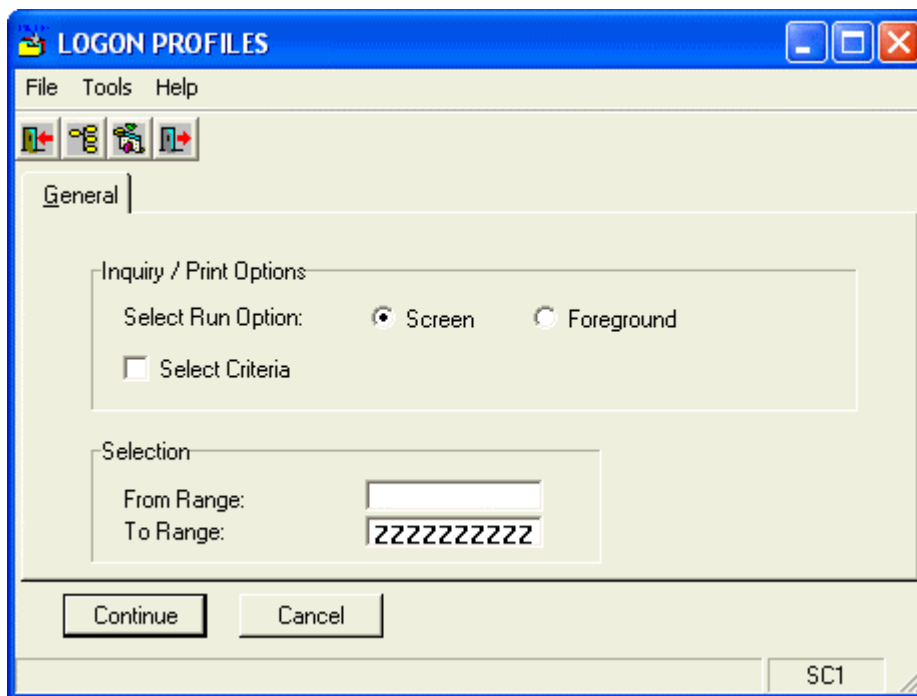
## 2.5 User Profile Inquiry / List

### *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.

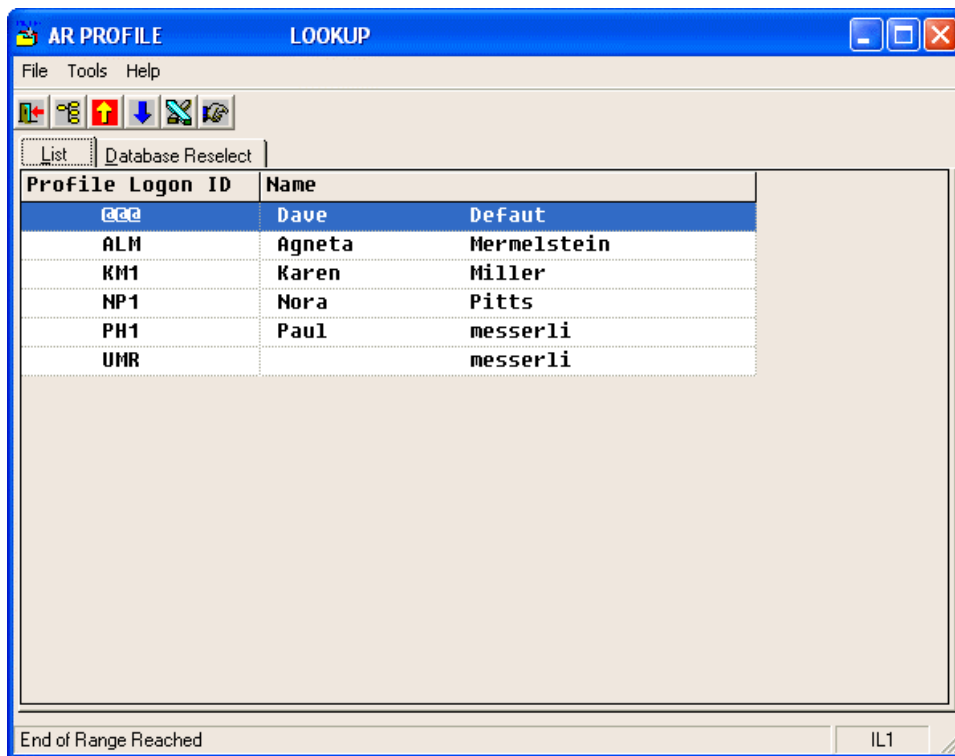
### To Range

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

## User Profiles Report Description

### User Profiles 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

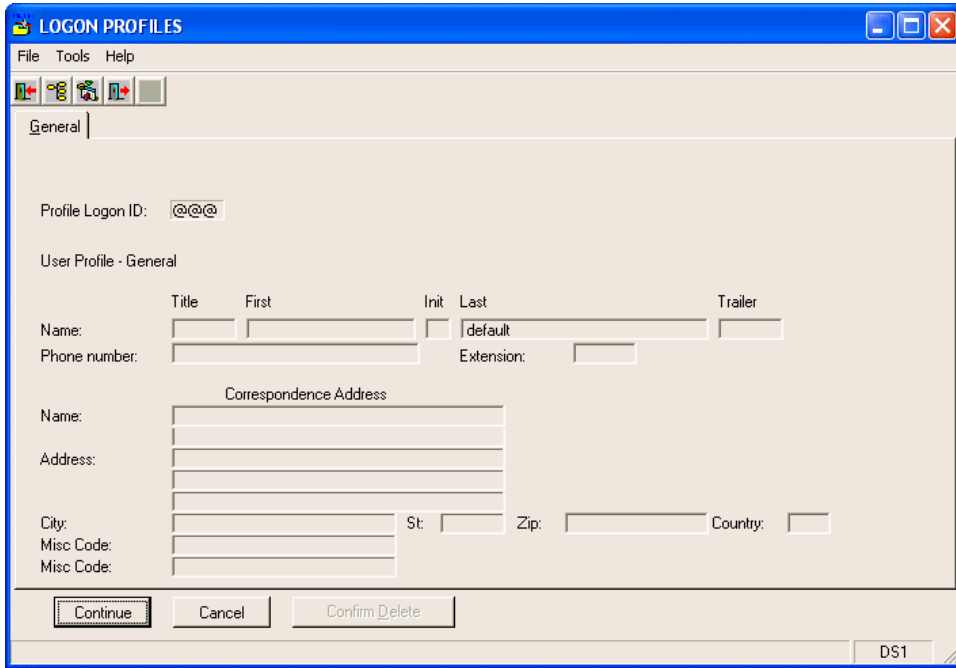
At the bottom of the window, the status bar displays "End of Range Reached" on the left and "IL1" on the right.

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:



**LOGON PROFILES**

File Tools Help

General

Profile Logon ID: @@@@

User Profile - General

Name: Title First Init Last Trailer

Phone number: Extension:

Correspondence Address

Name:

Address:

City: St: Zip: Country:

Misc Code:

Misc Code:

Continue Cancel Confirm Delete

DS1

User Profile Inquiry/List Detail Window (DS1)

## User Profiles 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.

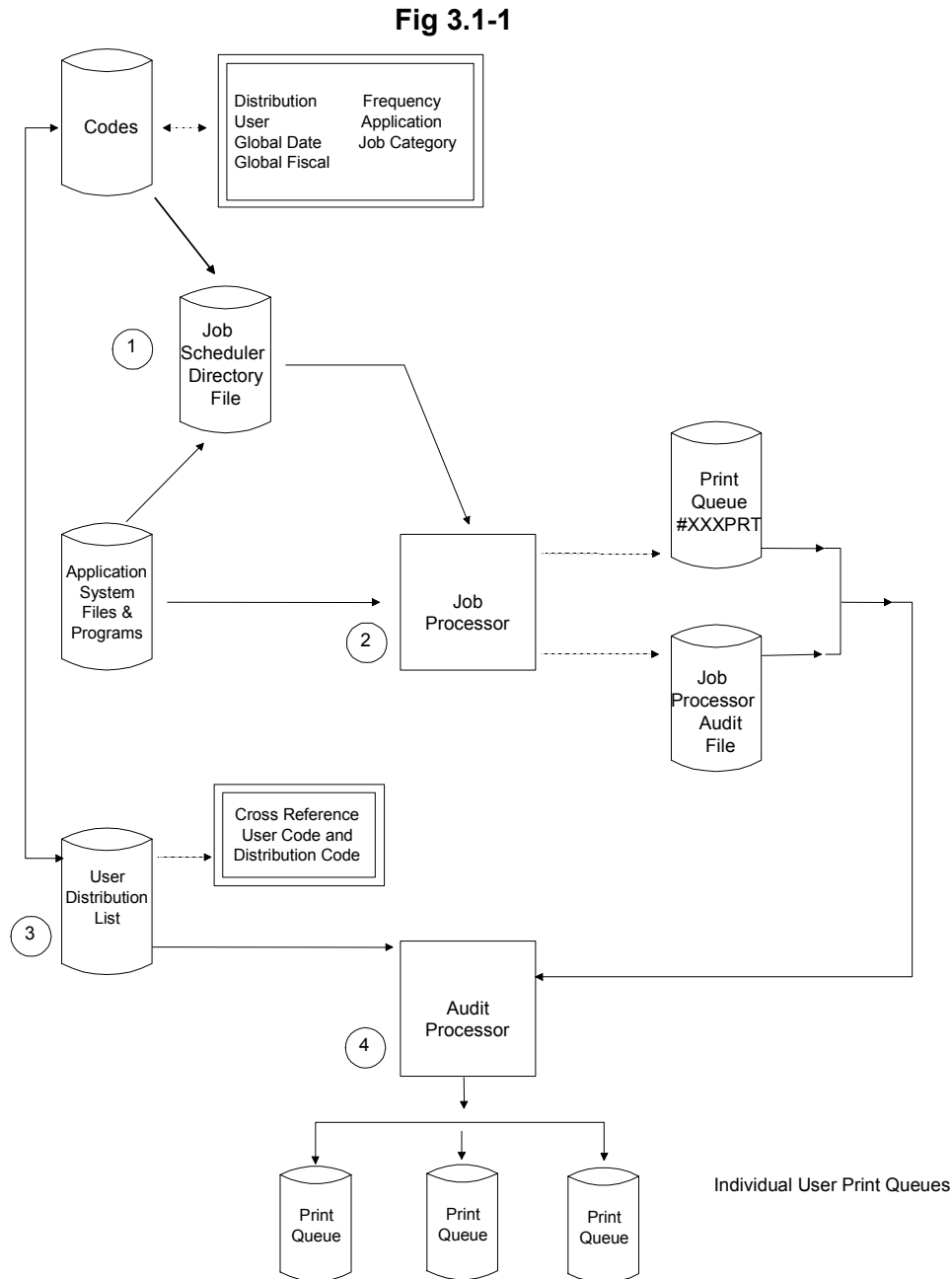
Job Processing can be used to run any of the following job types:

- ◆ 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.



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.).

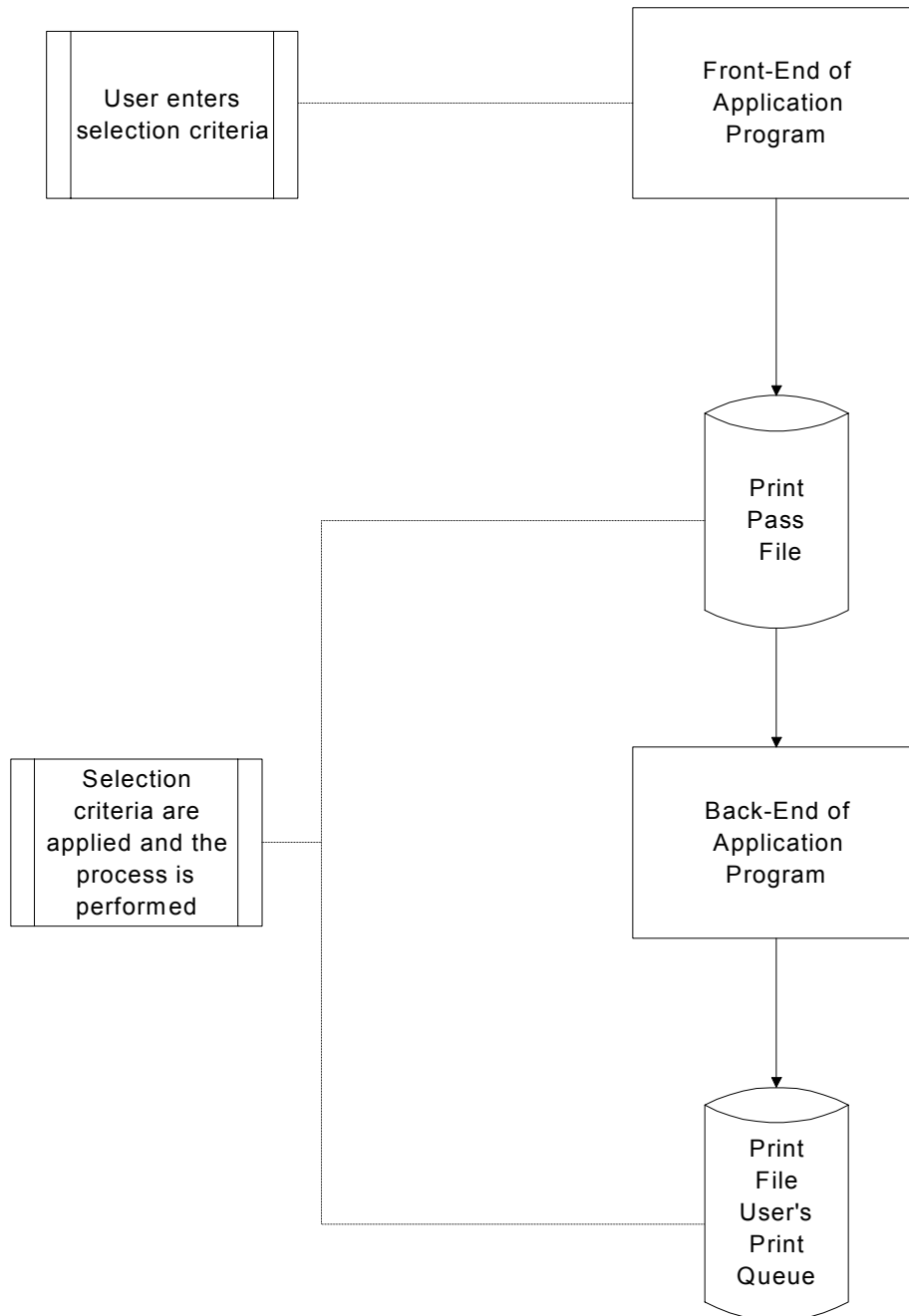
In addition this function produces the Process Audit report. This displays the following information for each user: the user's print library, the reports the user is to receive, the distribution code assigned to each report, the print files within which the reports are stored and the start and end time of report generation.

### *Other Programs*

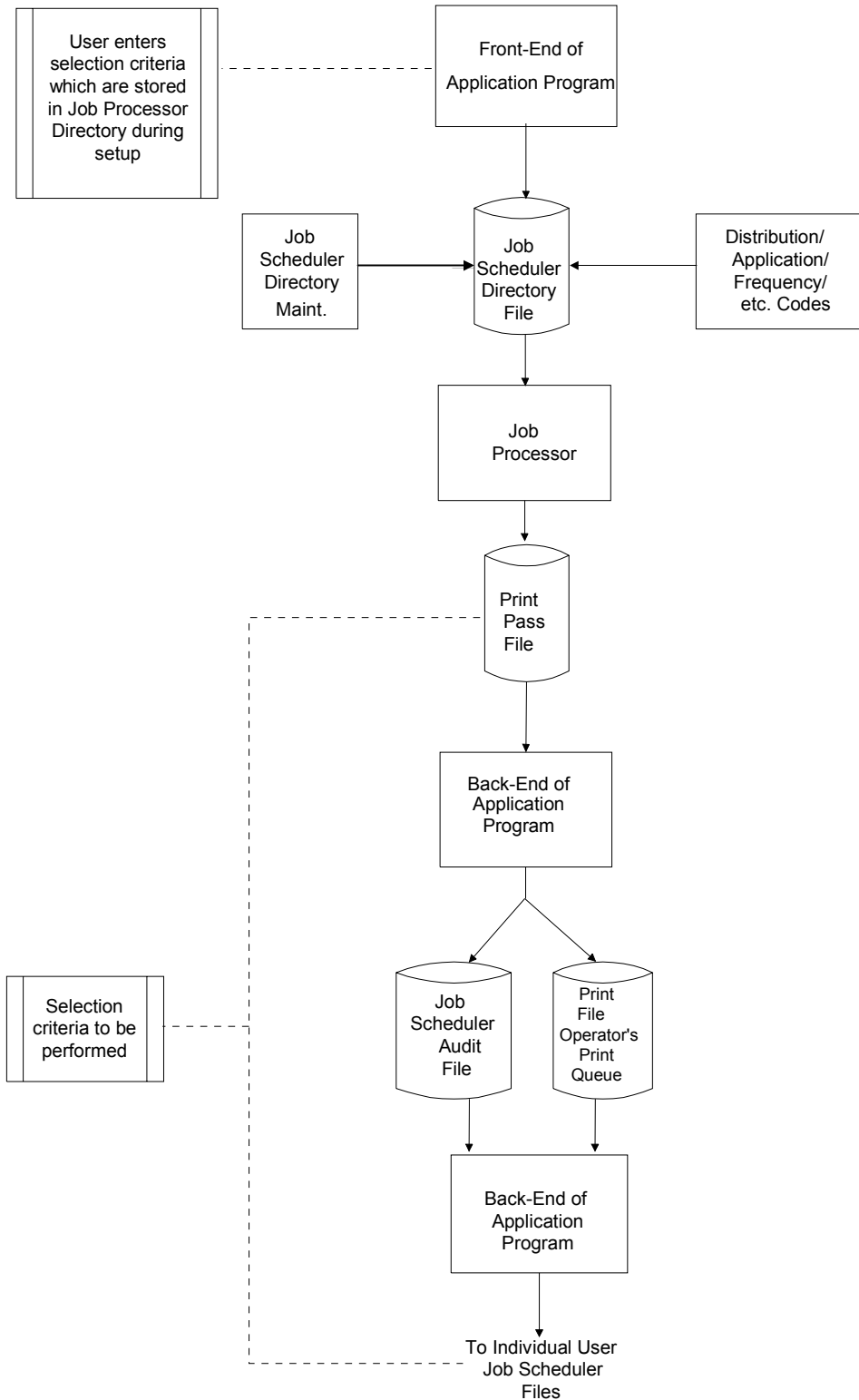
In addition to the major components described above, the Job Processing module includes many other functions. Several functions are used to create and maintain the various user-defined codes: distribution, frequency, application, job category and date codes. Inquiry/list functions are also available, 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.

<b>Table 3.1-1            Steps to Establish the Job Processor Environment</b>		
<b>Step No.</b>	<b>Description</b>	<b>Functions</b>
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

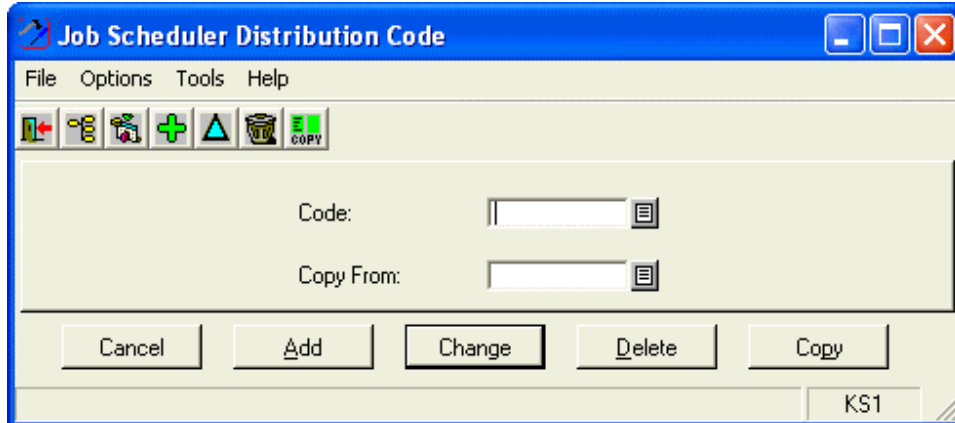
### *Introduction to Job Scheduler Code List Maintenance*

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-<sup>Mark</sup>) 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-<sup>Mark</sup> windows.

### Active

Enter the status of the code to maintain. This field appears on Q-<sup>Mark</sup> 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)

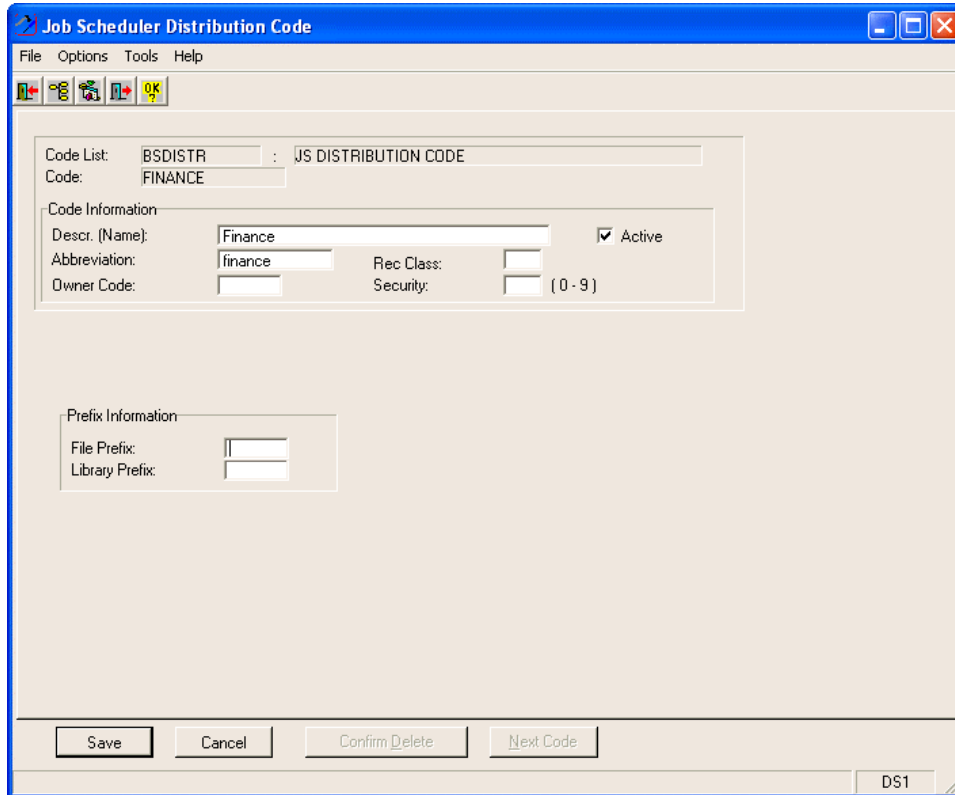
Enter a name up to thirty alphanumeric characters that describes the appropriate code. This field appears on Q-<sup>Mark</sup> windows.

### 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-<sup>Mark</sup> 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.

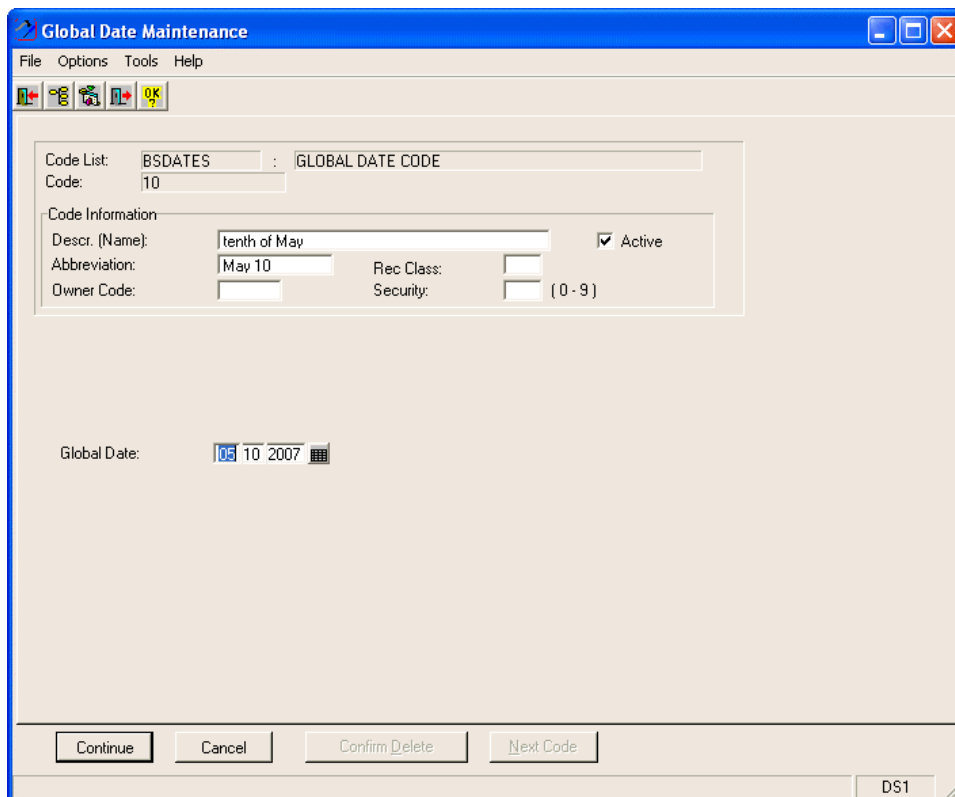


## Global Date Code Maintenance

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

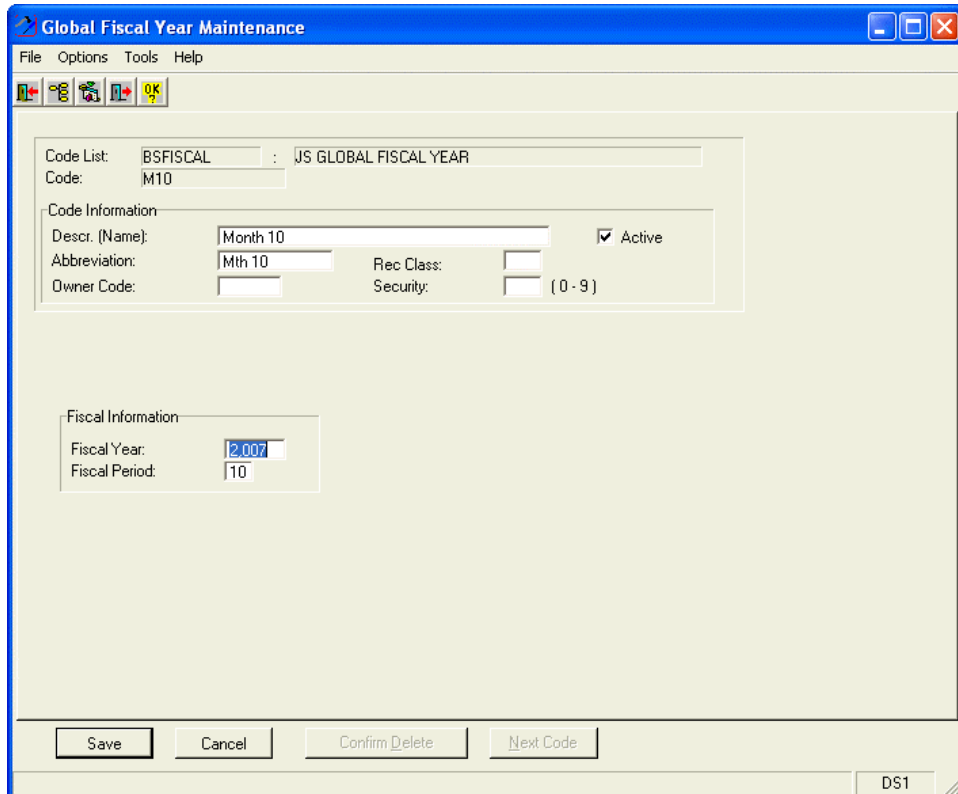
Enter the date to associate with this code. You can use the calendar icon next to the date field to select a date.

## 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

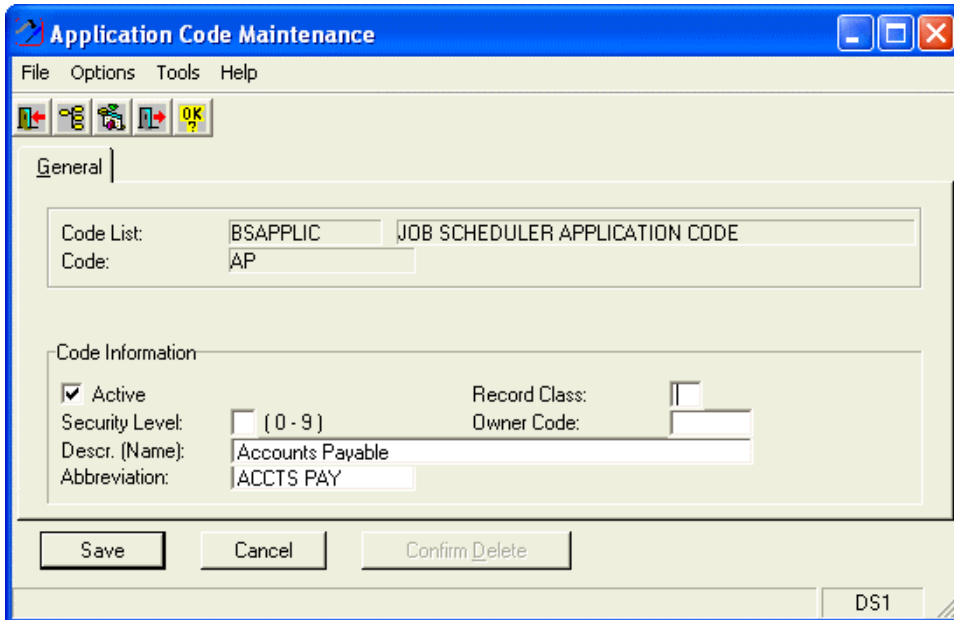
Enter the fiscal year to associate with this code.

### 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.

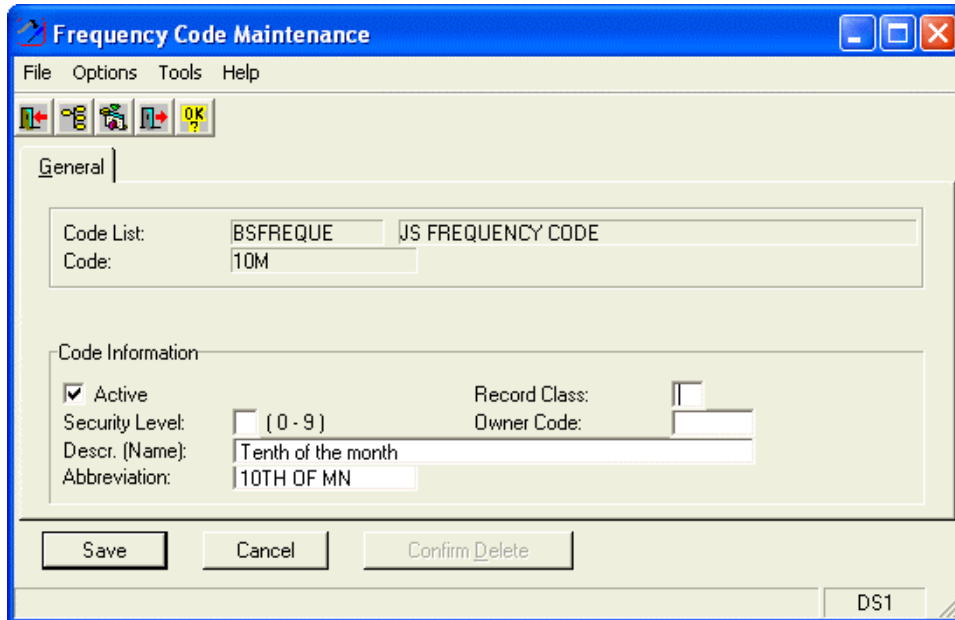


Application Code Maintenance Window (DS1)

## 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).

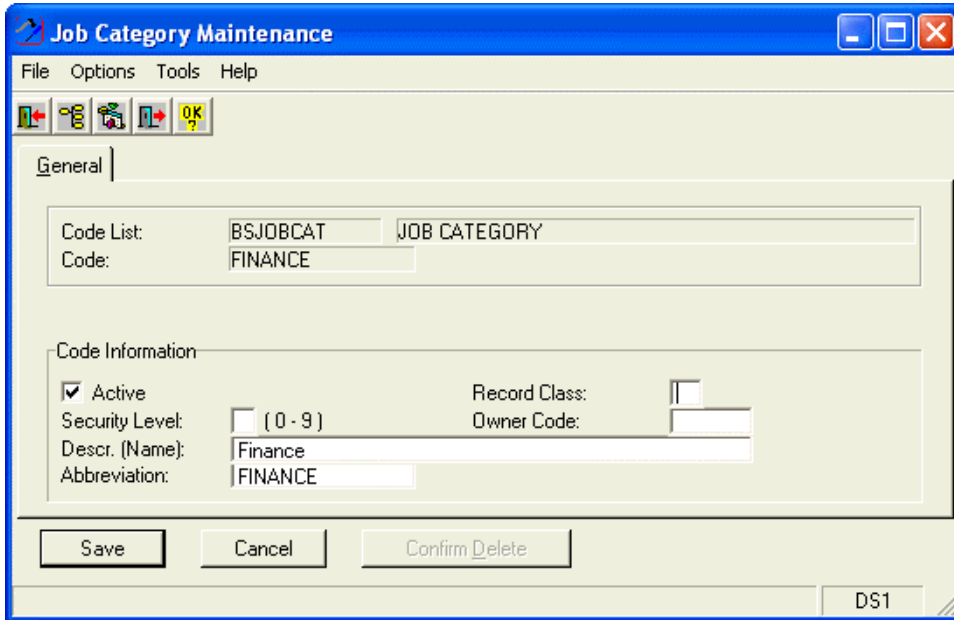
When a job is added to the Job Scheduler Directory, it is assigned a frequency code. This code can be generic (i.e., one of the system options) or user-defined. (e.g., D for daily, M for monthly etc.). When initiating the Job Processor, you can select the jobs to run based on the frequency code. For instance, you can elect to run only those jobs with a frequency code of M.



Frequency Code Maintenance Window (DS1)

## Job Category Code Maintenance

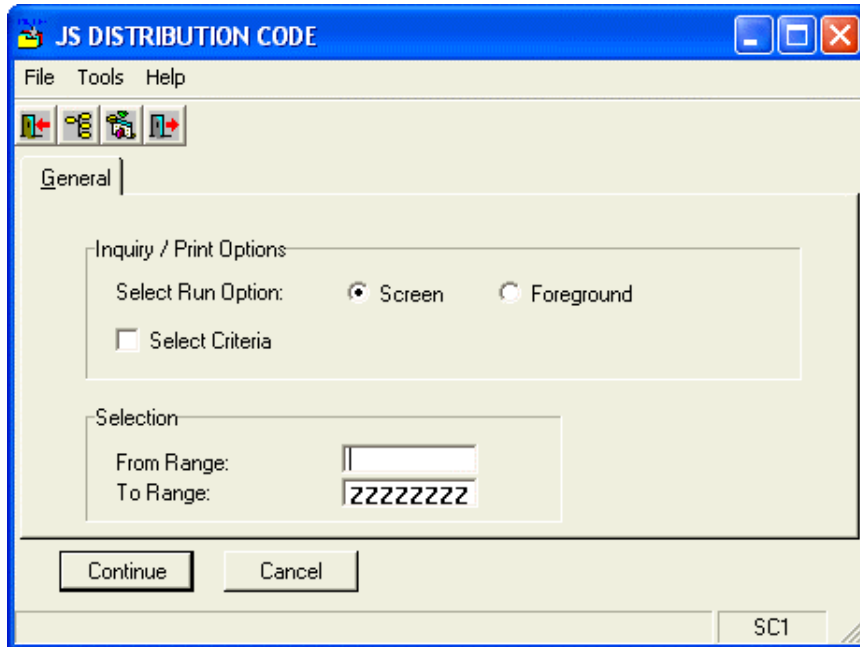
Job category codes are similar to application codes. These codes provide an additional means of classifying jobs.



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.

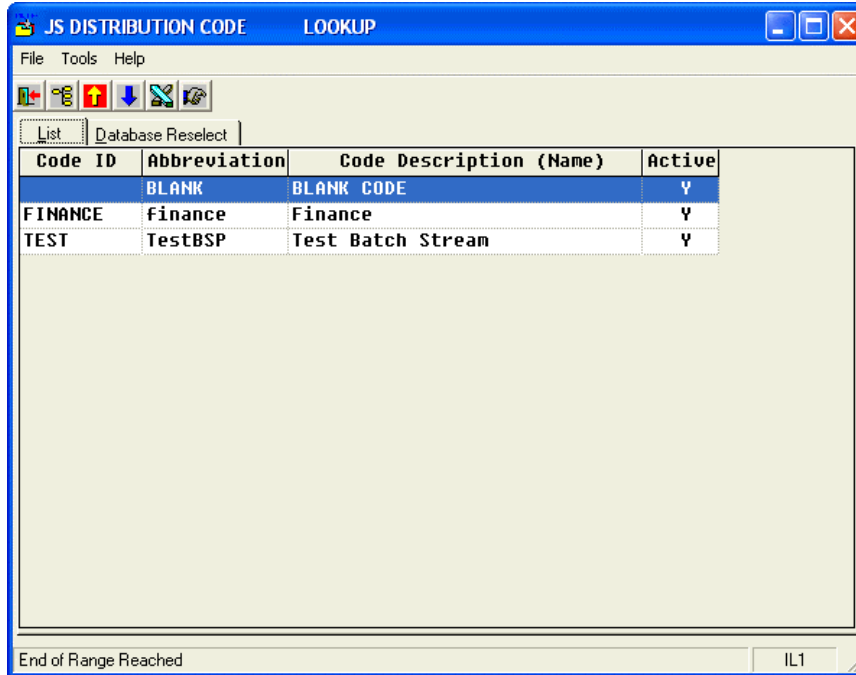
Do not allow modification of the selection criteria.

*Default:*

The remaining fields differ, depending on the List function selected.


## Job Processor Codes Screen Option

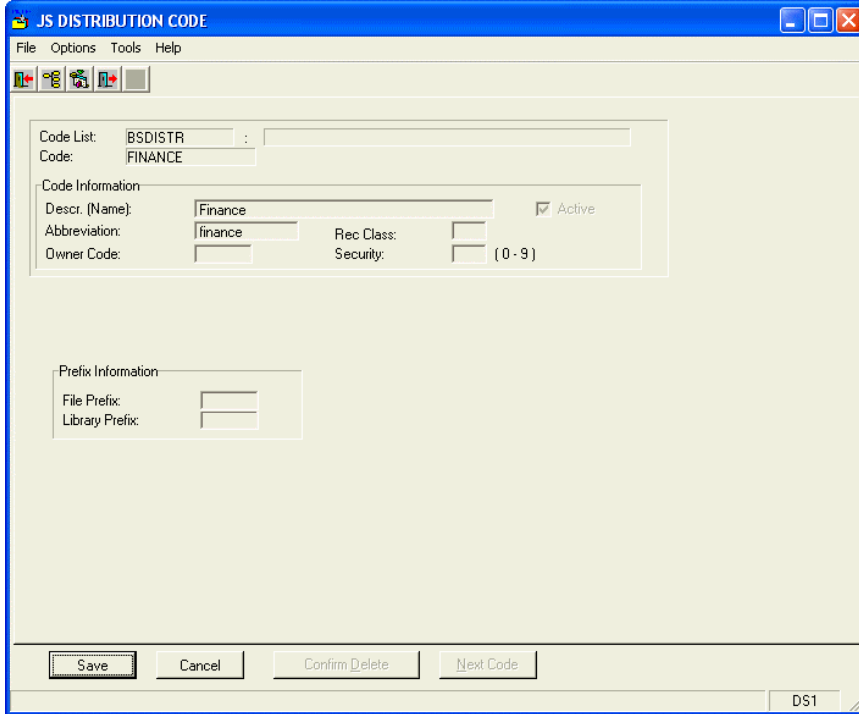
When the Screen option is selected, an inquiry window displays.



**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
  - Abbreviation: finance
  - Owner Code: [ ]
  - Rec Class: [ ]
  - Security: [ ] (0-9)
  - Active:
- Prefix Information:
  - File Prefix: [ ]
  - Library Prefix: [ ]

Buttons at the bottom include Save, Cancel, Confirm Delete, and Next Code. A status bar at the bottom right shows "DS1".

Inquiry Window Detail Display (DS1)

## Job Processor Codes 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

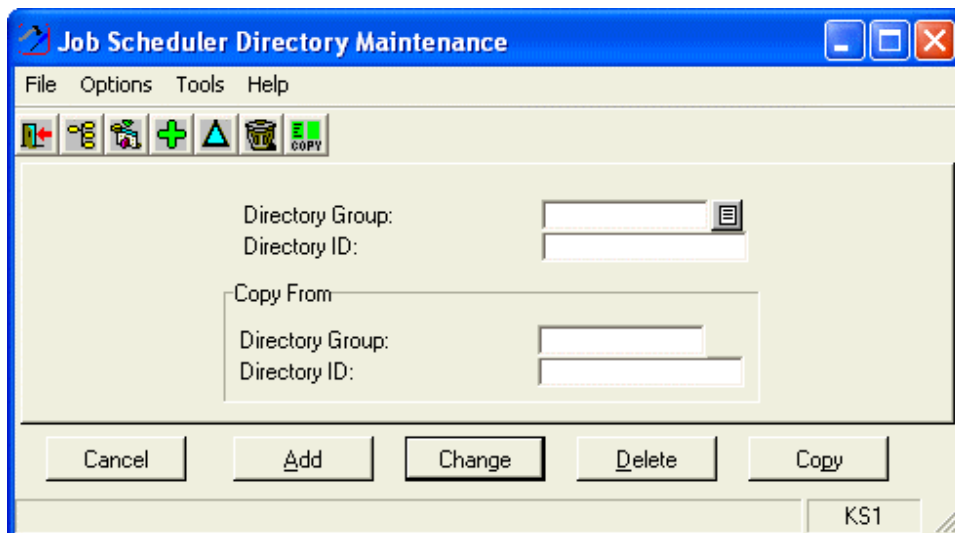
### *Introduction to Job Scheduler Directory Maintenance*

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.

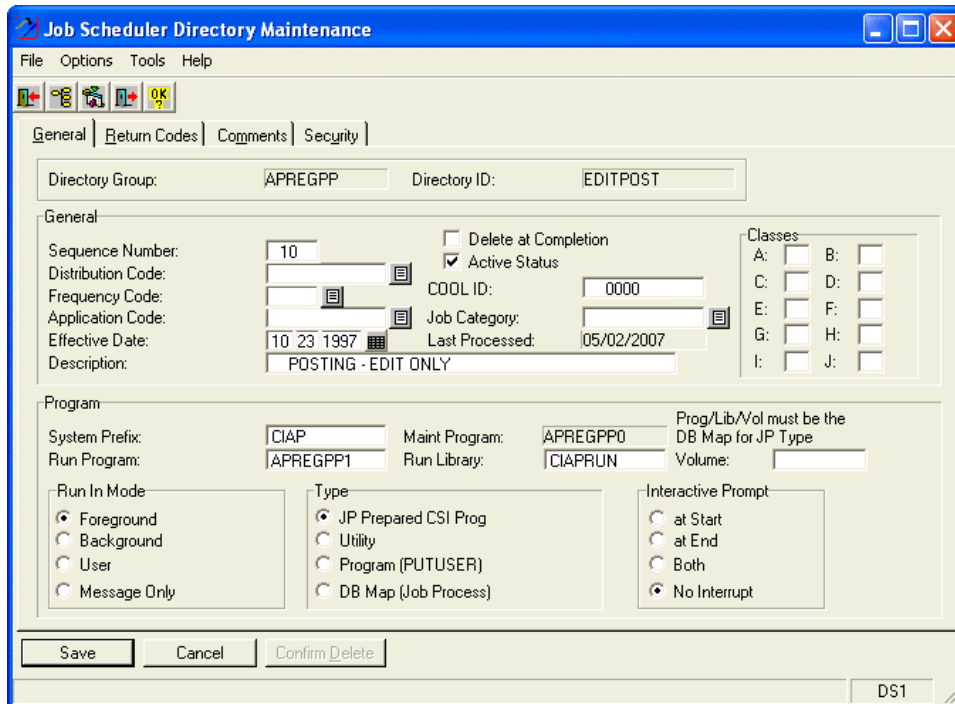
*Note:* When adding a job to the directory, do not use one of the directory groups associated with a Job Processor-Prepared Computron program.

## Directory ID (Q-<sup>Mark</sup>)

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-<sup>Mark</sup>), and directory ID (Q-<sup>Mark</sup>), 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)



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-<sup>Mark</sup>)

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-<sup>Mark</sup>)

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-<sup>Mark</sup>. 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 n<sup>th</sup> 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.

<b>Table 3.3-1 Frequency Codes — System Options</b>	
<b>Code</b>	<b>Description</b>
#W $n$	$n$ = day of week (#W1 = Sunday, #WN = Sunday)
# $nn$	$nn$ = day of month (#32 = last day of month)
* $w$ $n$	$w$ = week of month $n$ = day of week (*67 = Last Saturday of month)

*Examples:*

<b>Code</b>	<b>Description</b>
#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 30<sup>th</sup> day of month), \*54 (5<sup>th</sup> 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.

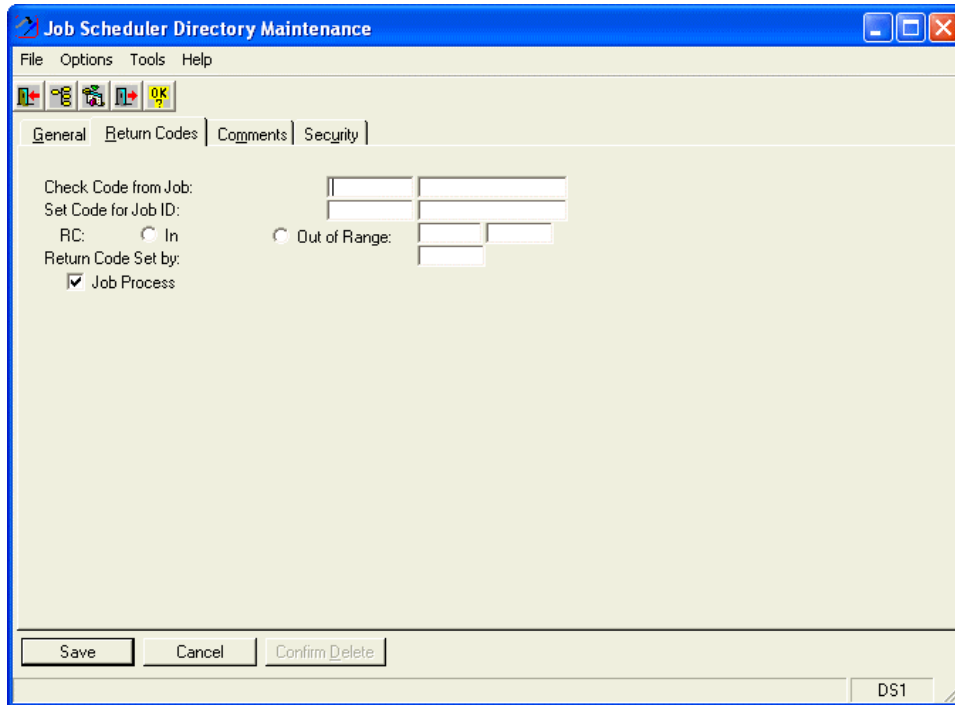
*Default:* No Interrupt.

☞ *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.

Table 3.3-2 Return Codes — System-Defined	
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.

<b>Table 3.3-2 Return Codes — System-Defined</b>	
<b>Return Code</b>	<b>Description</b>
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:

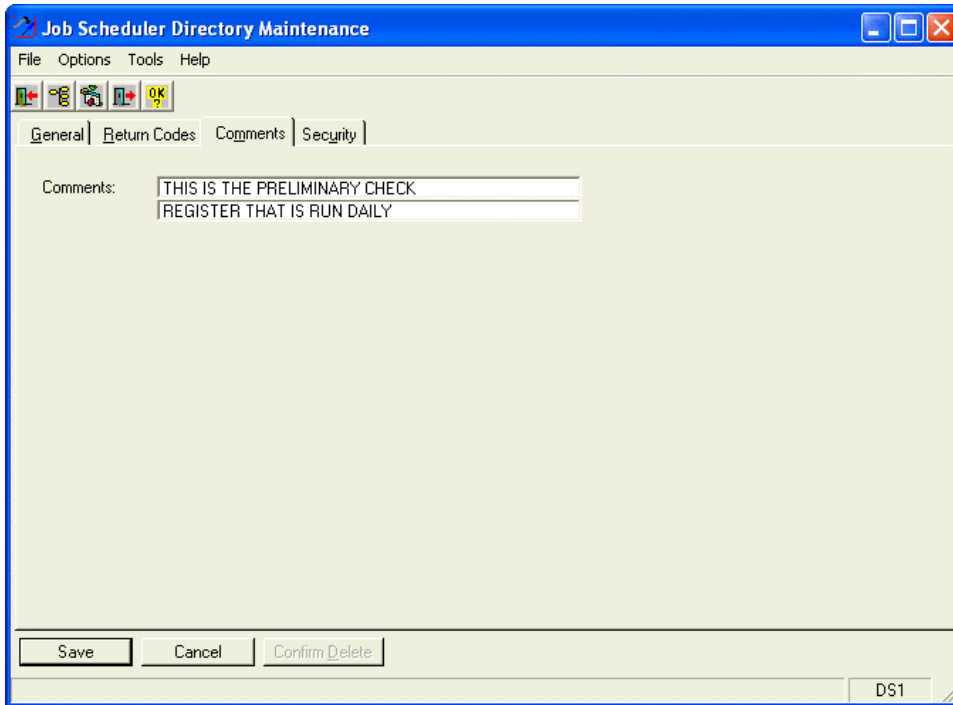
- Write the user-defined return code to the Return Code file.
- Do not write the user-defined return code to the Return Code file.

*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:



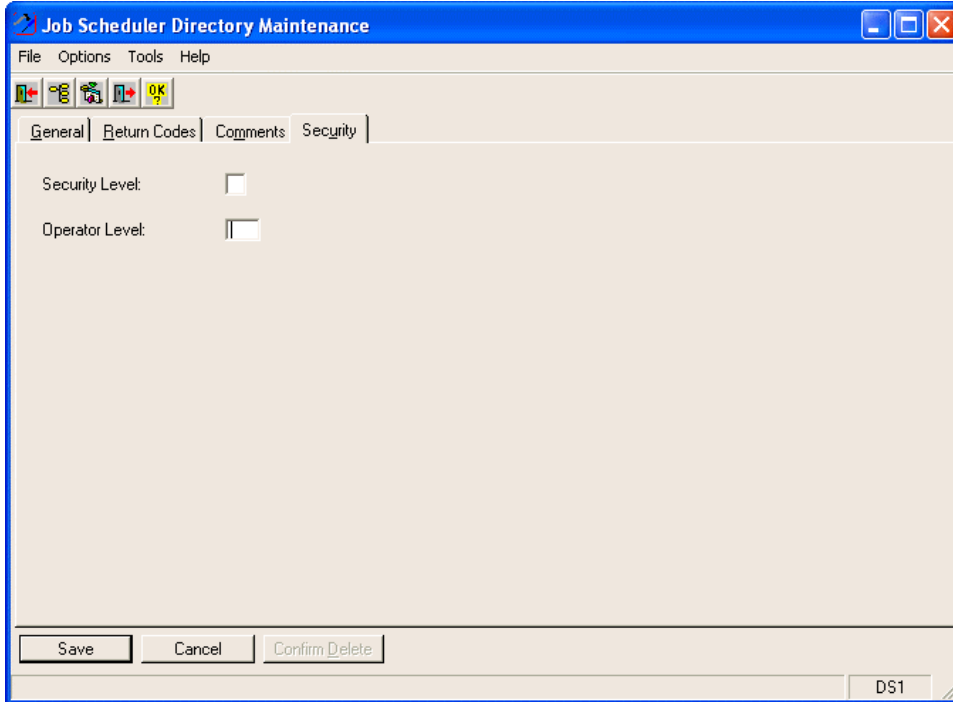
**Job Scheduler Directory Maintenance Window (DS1) – Comments**

### 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**

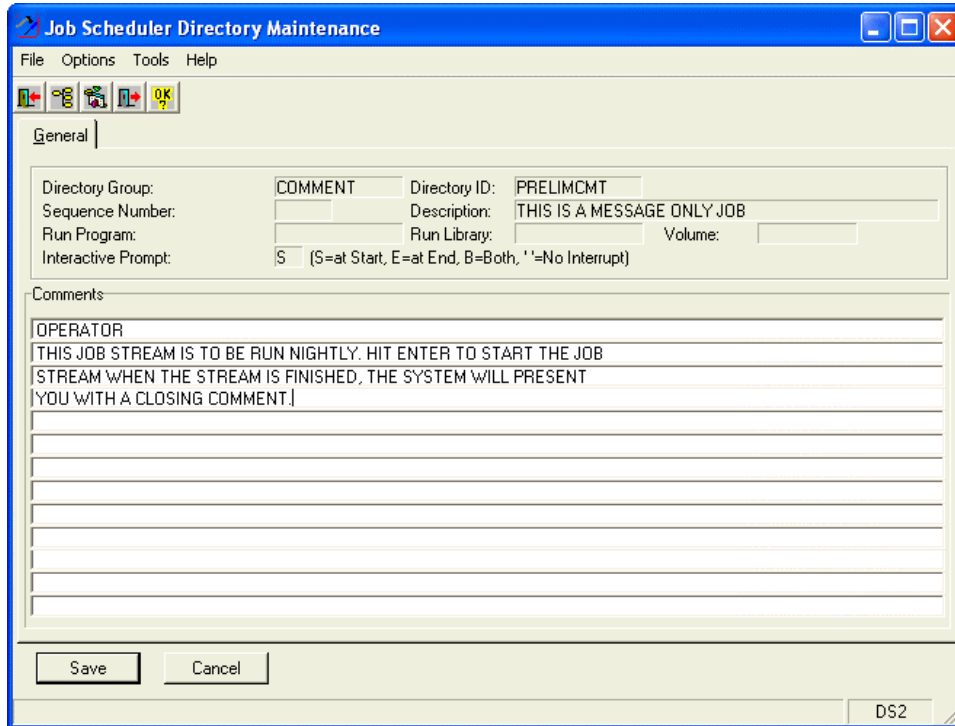
This field is not in use.

### **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

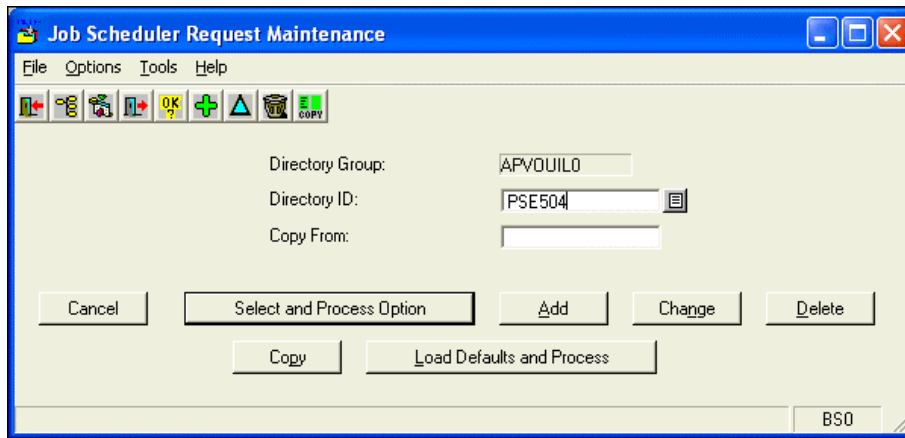
Use this area to enter the message to display on the operator's window at run-time.

## 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.

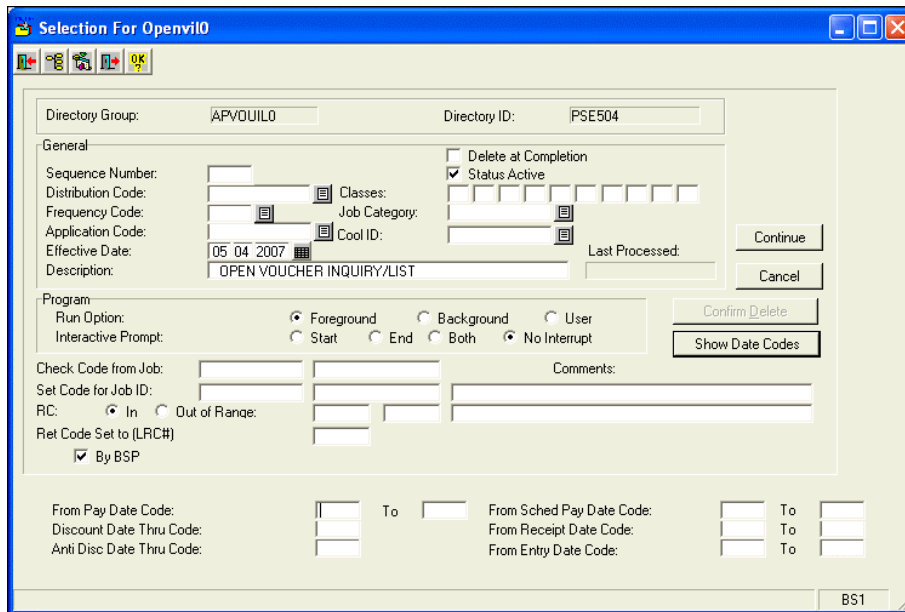
The first window — Job Scheduler Maintenance Window (BS0) prompts for a directory ID and the selection of a maintenance option (Add, Change, Copy, or Delete). The directory group is automatically assigned by the program and cannot be modified.

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<sup>Mark</sup> 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.

<b>Table 3.3-3 Date Codes — System Options</b>	
<b>Code</b>	<b>Description</b>
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)

<b>Table 3.3-3 Date Codes — System Options</b>	
<b>Code</b>	<b>Description</b>
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 Return Codes 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:

<b>Job 1 (Backup)</b>	
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.

<b>Job 2 (Posting)</b>	
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.

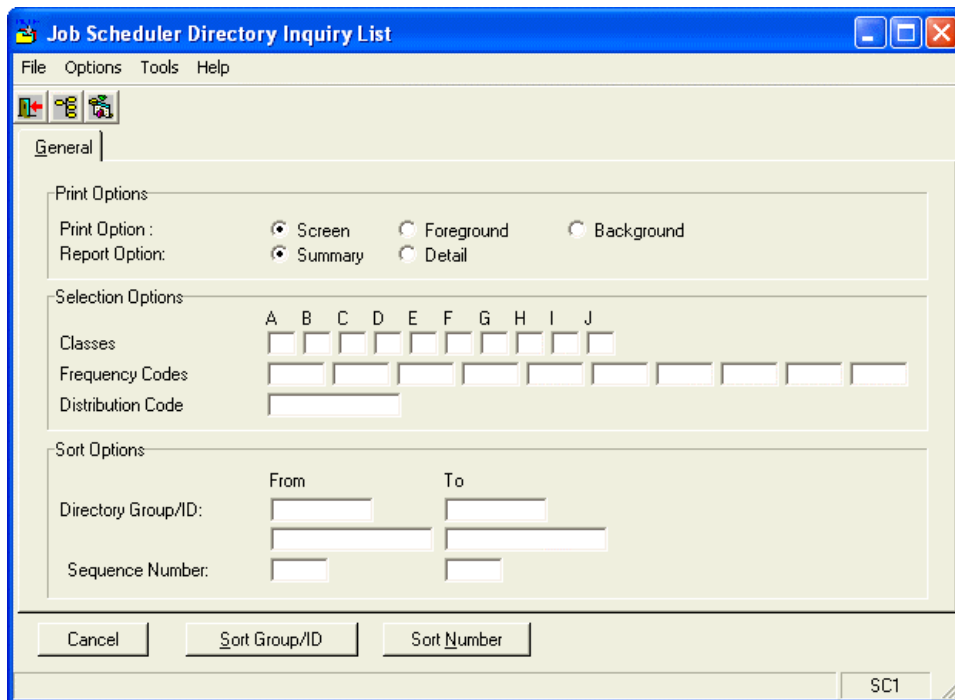
## 3.4 Job Scheduler Directory Inquiry/List

### *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
<b>Sort Group/ID</b>	Display the jobs in order by directory group, and within directory group, by directory ID.
<b>Sort Number</b>	Display the jobs in order by sequence number.

# Job Scheduler Report Description

## Job Scheduler 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	FURCRPB	CIAPRUN		
0	APVOUIL0	TEST	Y		Open Voucher Inquiry/List	OPENVILL1	CIAPRUN		
0	FARPT	PROPI	Y		Property Listing Report	FAREPRT1	CIAPRUN		
0	FARPT	DISP1	Y		DISPOSALS REPORT	FAREPRT1	CIAPRUN		
0	FARPT	EXPR1	Y		DEPRECIATION EXPENSE REPORT	FAREPRT1	CIAPRUN		
0	FARPT	SNAP	Y		Data Extraction for Above Rpts	FAREPRT1	CIAPRUN		
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	CIAPRUN		
10	APREGPP	APREGPSE	Y		POSTING - EDIT ONLY	APREGPP1	CIAPRUN		
20	APREGPP	UPDATEPOST	Y		Posting Update	APREGPP1	CIAPRUN		
40	APCKPP	PRELIMCK	Y		PRELIMINARY CHECK REGISTE	WRITICKO	CIAPRUN		
50	FABATREL	BATCHEDIT	Y		Batch Transaction Release	FABATRL1	CIAPRUN		
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	CIAPRUN		
85	FAPOST	FAGLEPOST	Y		FA TO G/L EDIT POSTING	GLPOST1	CIAPRUN		
95	FAPOST	FAGLPOST	Y		FA TO GENERAL LEDGER POSTING	GLPOST1	CIAPRUN		

### Job Scheduler Directory Report (Detail)

```

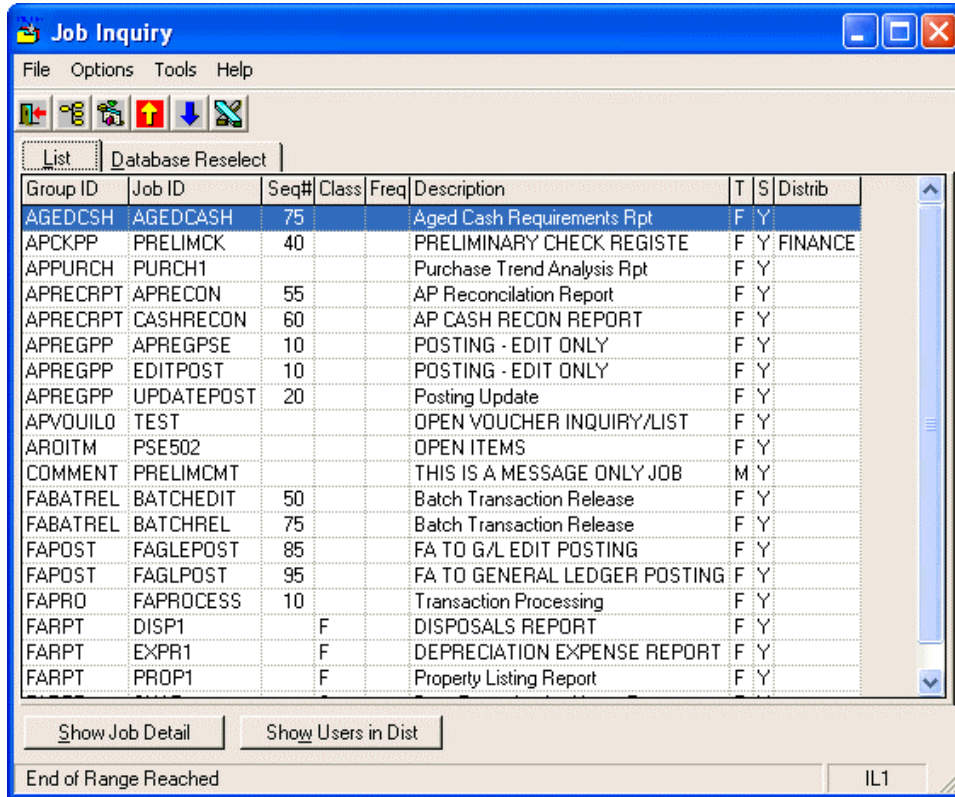
*   Computron Software, LLC.   ***** Wed May 02, 2007 10:42 AM *
*
*                               JOB SCHEDULER DIRECTORY DETAIL REPORT
*                               SORTED BY: DIRECTORY GROUP/ID
*
*----- Page 1 *

```

D I R E C T O R Y		SEQ	DESCRIPTION / COMMENTS	EFFECTIVE/ ST DISTRIB	OL/ 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	9	AGEDCSH1 FOREGROUND CIAPRUN NONE OTHER
APCKPP	PRELIMCK	40	PRELIMINARY CHECK REGISTE	Y			10/27/97 05/01/07	9	9	WRITICKO FOREGROUND CIAPRUN START OTHER
APPURCH	PURCH1		Purchase Trend Analysis Rpt	Y			09/26/06	9	9	PURCRPB FOREGROUND CIAPRUN NONE OTHER
APRECRPT	APRECON	55	AP Reconciliation Report	Y			05/01/07	9	9	APRECON1 FOREGROUND CIAPRUN NONE OTHER
APRECRPT	CASHRECON	60	AP CASH RECON REPORT	Y			05/01/07	9	9	APRECON1 FOREGROUND CIAPRUN NONE OTHER
APREGPP	APREGPSE	10	POSTING - EDIT ONLY	Y			05/01/07 05/01/07	9	9	APREGPP1 FOREGROUND CIAPRUN NONE OTHER
APREGPP	EDITPOST	10	POSTING - EDIT ONLY	Y			10/23/97 05/01/07	9	9	APREGPP1 FOREGROUND CIAPRUN NONE OTHER
APREGPP	UPDATEPOST	20	Posting Update	Y			10/23/97 05/01/07	9	9	APREGPP1 FOREGROUND CIAPRUN NONE OTHER
APVOUIL0	TEST	0	Open Voucher Inquiry/List	Y			08/30/06 08/30/06	9	9	OPENVILL1 FOREGROUND CIAPRUN NONE OTHER
COMMENT	PRELIMCMT	0	THIS IS A MESSAGE ONLY JOB	Y			05/02/07	0	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	9	FABATRL1 FOREGROUND CIAPRUN NONE OTHER
FABATREL	BATCHREL	75	Batch Transaction Release	Y			10/24/97 05/01/07	9	9	FABATRL1 FOREGROUND CIAPRUN NONE OTHER

## Job Scheduler Screen Option

When you request the Job Scheduler Directory I/L function to produce the information to Screen, the following IL1 window displays.

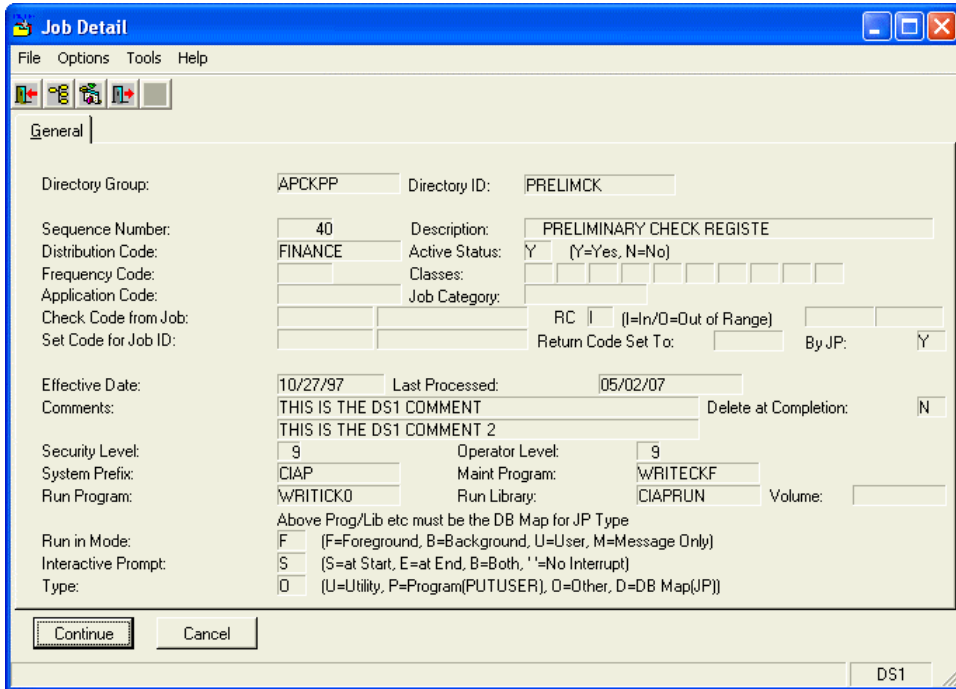


The screenshot shows a window titled "Job Inquiry" with a menu bar (File, Options, Tools, Help) and a toolbar. Below the toolbar are buttons for "List" and "Database Reselect". The main area contains a table with the following columns: Group ID, Job ID, Seq#, Class, Freq, Description, T, S, and Distrib. The table lists various jobs such as AGEDCASH, PRELIMCK, PURCH1, APRECON, CASHRECON, APREGPSE, EDITPOST, UPDATEPOST, TEST, PSE502, PRELIMCMT, BATCHEDIT, BATCHREL, FAGLEPOST, FAGLPOST, FAPROCESS, DISP1, EXPR1, and PROP1. 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	
APVOUILO	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)

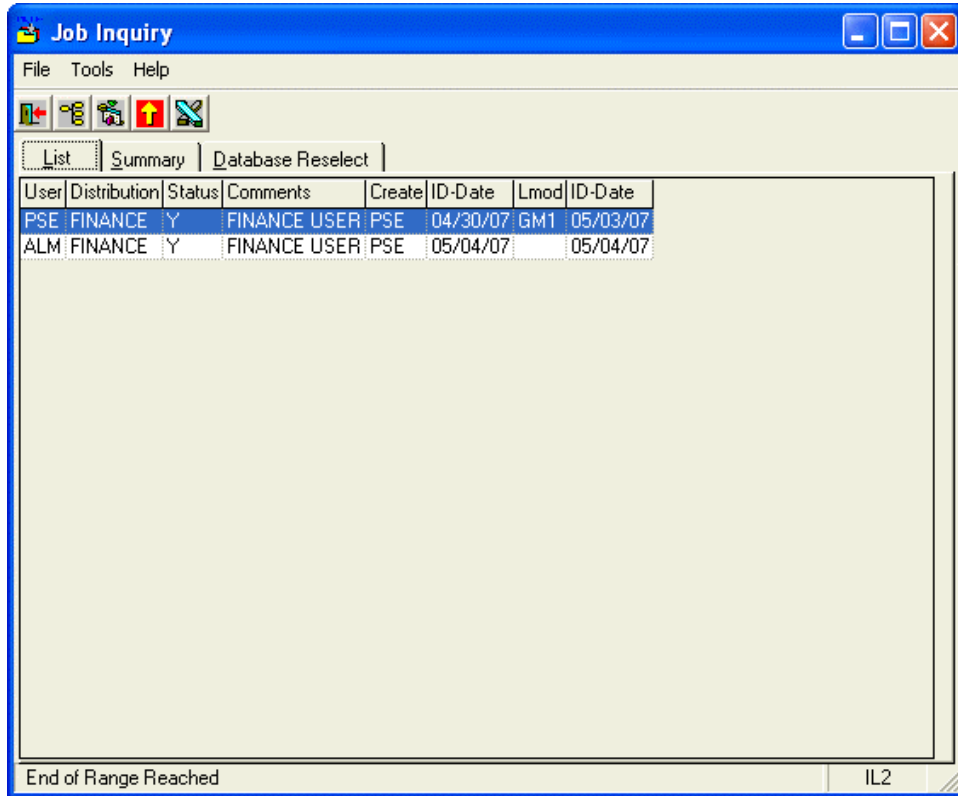
Type: O (U=Utility, P=Program(PUTUSER), O=Other, D=DB Map(JP))

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

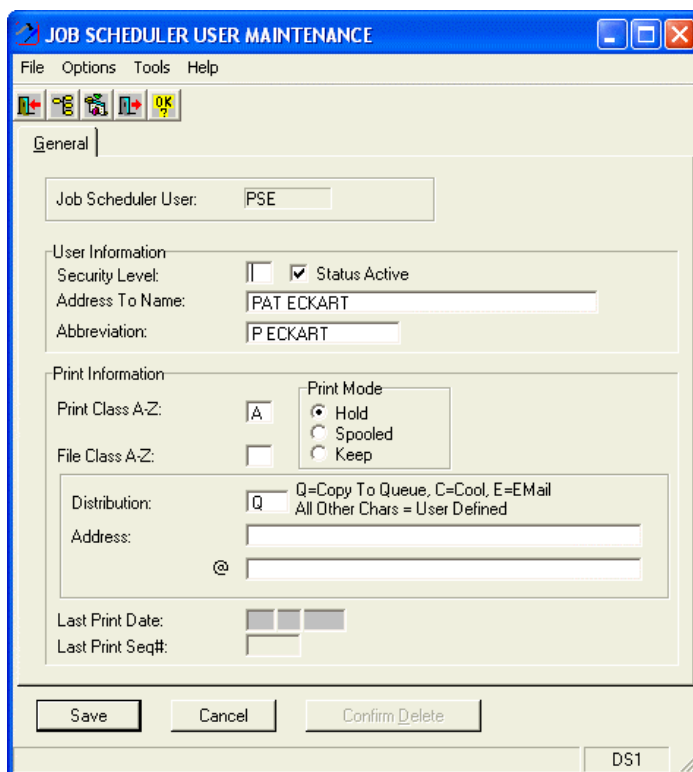
### *Introduction to User Code Maintenance*

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-<sup>Mark</sup>), 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-<sup>Mark</sup> 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-<sup>Mark</sup> 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-<sup>Mark</sup> 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

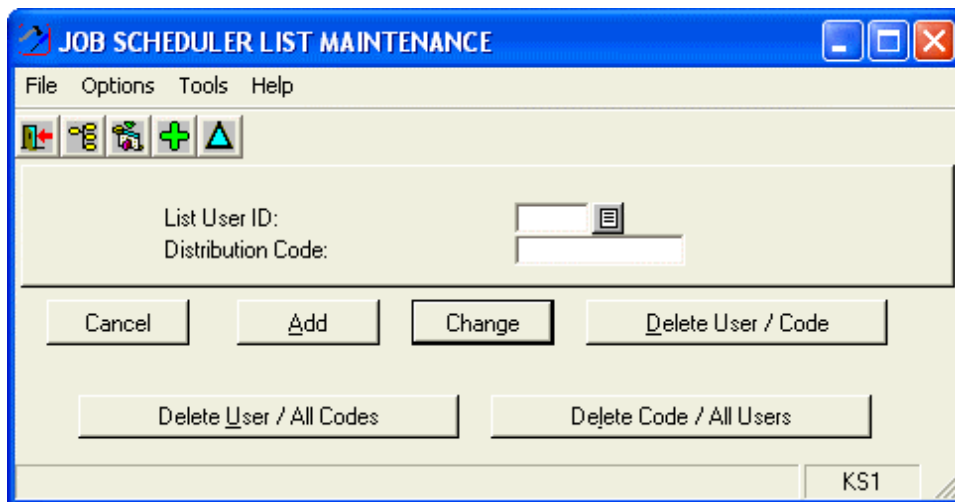
### *Introduction to User Distribution List Maintenance*

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)**

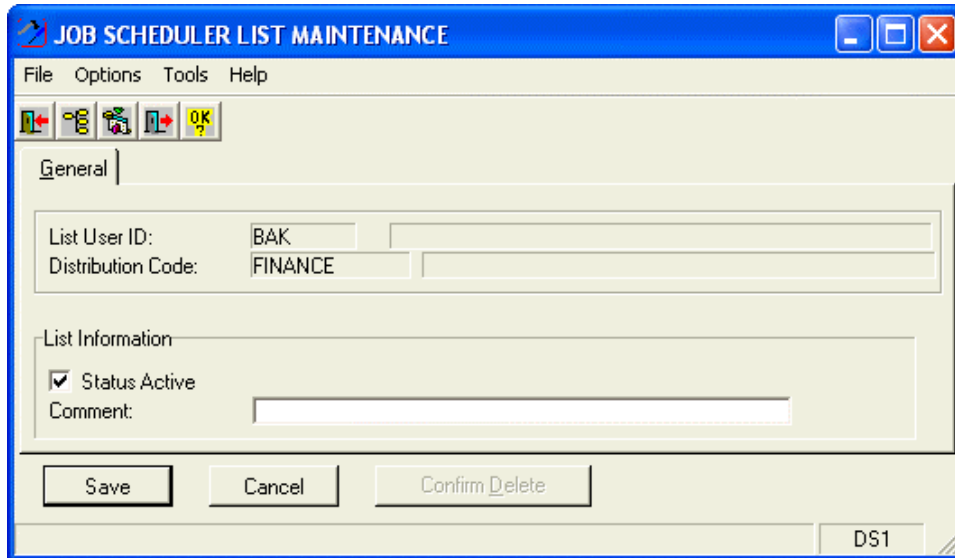
Button	Function
<b>Delete User / Code</b>	Delete a user code from a distribution code.
<b>Delete User / All Codes</b>	Delete a user code from all distribution codes. (Leave the Distribution Code field blank)
<b>Delete Code / All Users</b>	Delete a distribution code from all user codes. (Leave the List User ID field blank.)

From the User Distribution List Maintenance Window (KS1) enter the appropriate list user ID (Q<sub>Mark</sub>), 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-<sup>Mark</sup> 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-<sup>Mark</sup> windows, and on the User Distribution report.

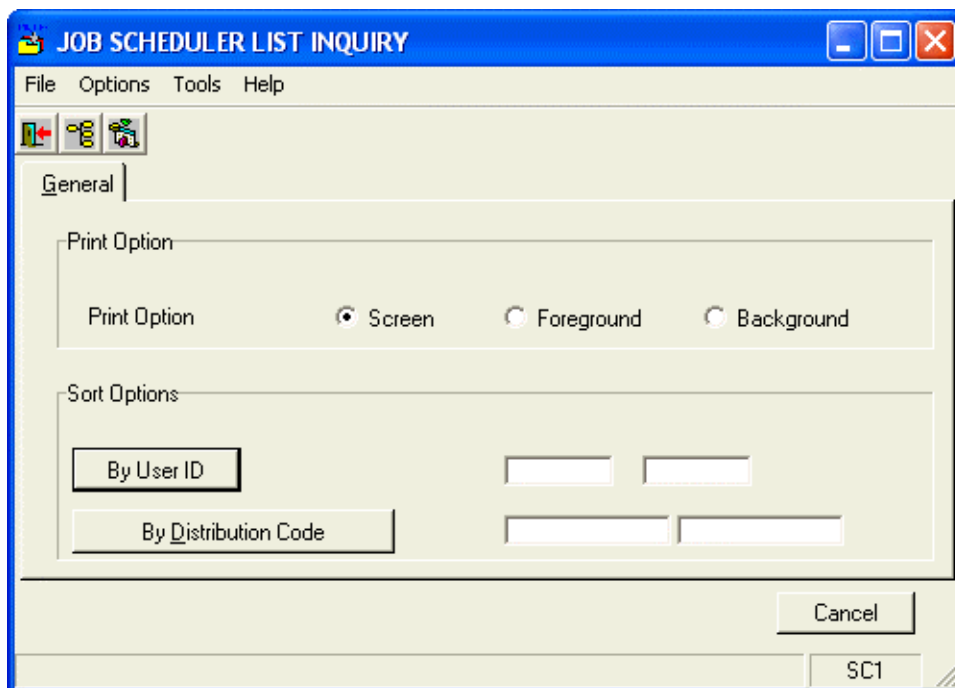
## 3.7 User Distribution List Inquiry / List

### *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:

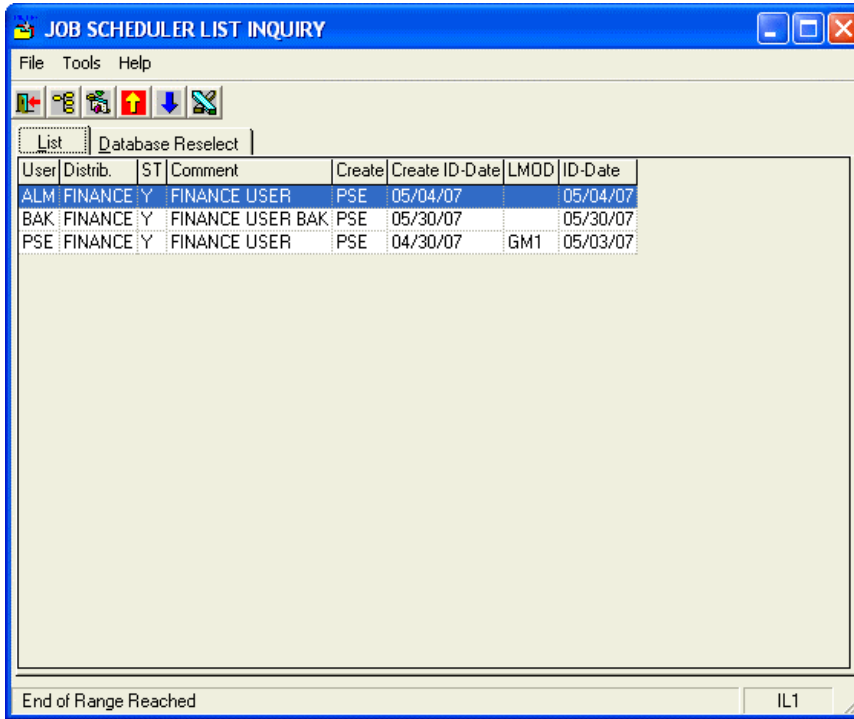
<b>Button</b>	<b>Function</b>
<b>By User ID</b>	Display the cross-reference entries in order by user ID.
<b>By Distribution Code</b>	Display the cross-reference entries in order by distribution code.

## *User Distribution List Report Options*

### **User Distribution List 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)



User Distribution List Window (IL1)

## User Distribution List 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





## 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., 23<sup>rd</sup> day of the month) and the processing date is October 23<sup>rd</sup>, 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 23<sup>rd</sup>, the date code is interpreted as October 31<sup>st</sup>.

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 29<sup>th</sup>, 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 30<sup>th</sup>, 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-<sup>Mark</sup>)

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-<sup>Mark</sup>)

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
<b>Inquiry</b>	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).
<b>Process All Jobs</b>	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).
<b>Process All Jobs in Background</b>	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).
<b>Select Starting Job and Process</b>	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).
<b>Process Single Job</b>	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).

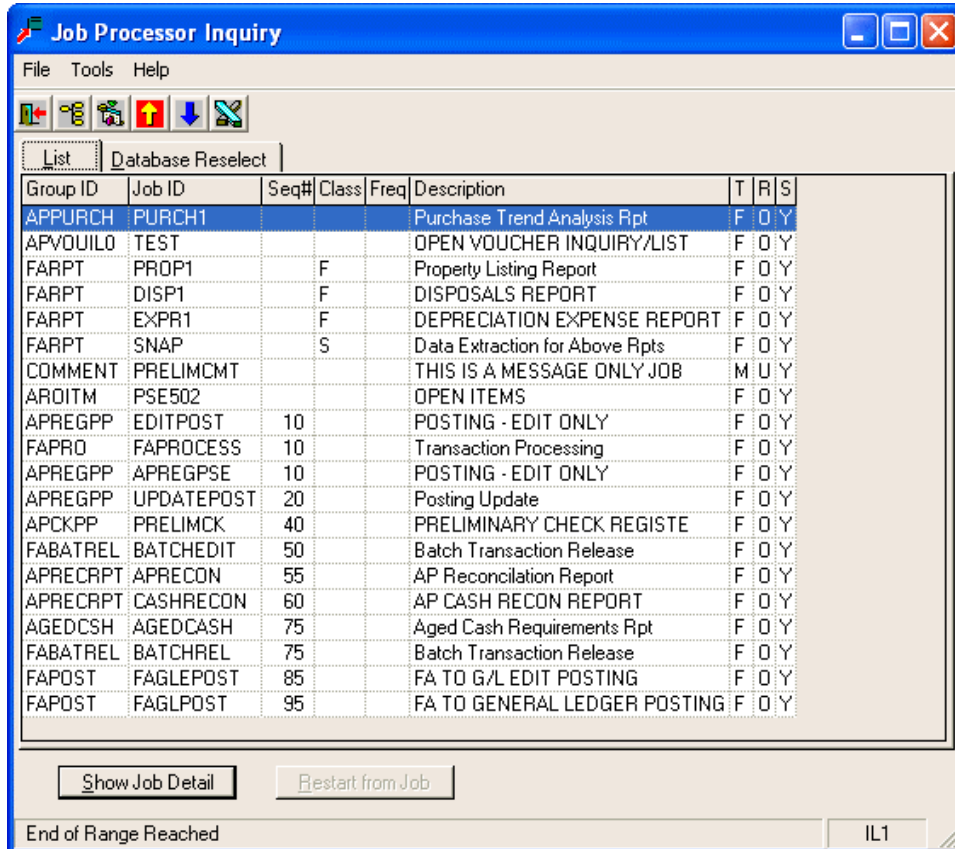
☞ *Note:* In all the above cases, the jobs are run in sequence number order.

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
APVOIULO	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

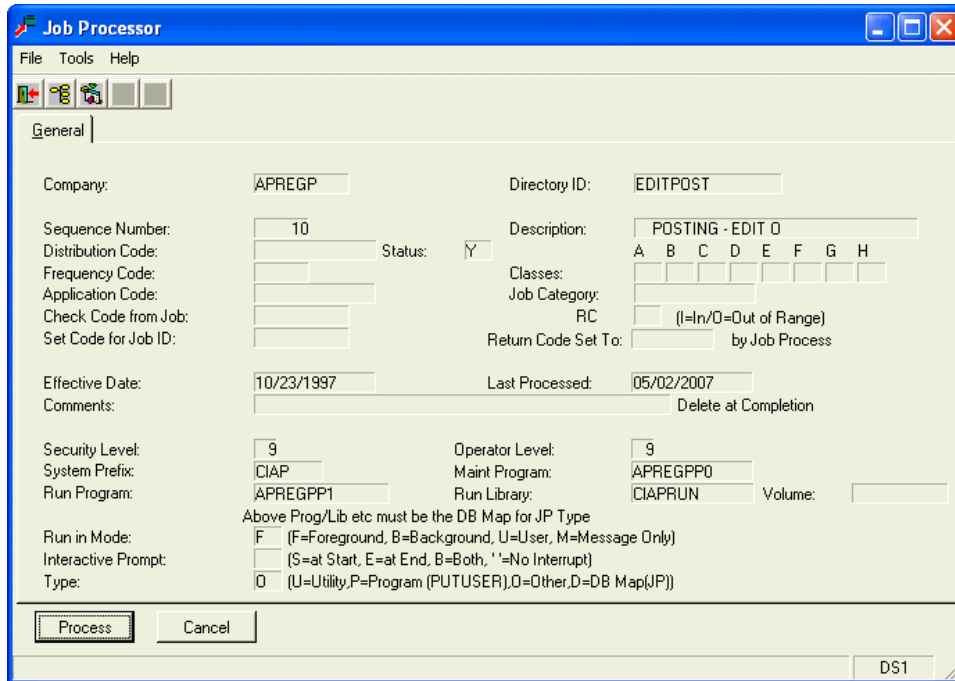
**Job Processor Inquiry Window (IL1)**

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 O
Distribution Code:		Status:	<input checked="" type="checkbox"/> Y
Frequency Code:		Classes:	A B C D E F G H
Application Code:		Job Category:	
Check Code from Job:		RC:	<input type="checkbox"/> (I=In/O=Out of Range)
Set Code for Job ID:		Return Code Set To:	by Job Process
Effective Date:	10/23/1997	Last Processed:	05/02/2007
Comments:	Delete at Completion		
Security Level:	9	Operator Level:	9
System Prefix:	CIAP	Maint Program:	APREGPP0
Run Program:	APREGPP1	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:	<input type="checkbox"/> (S=at Start, E=at End, B=Both, ''=No Interrupt)		
Type:	O (U=Utility, P=Program (PUTUSER), O=Other, D=DB Map(JP))		

Buttons: Process, Cancel

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 Job Processor produces the 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.



## 3.9 Audit Processor

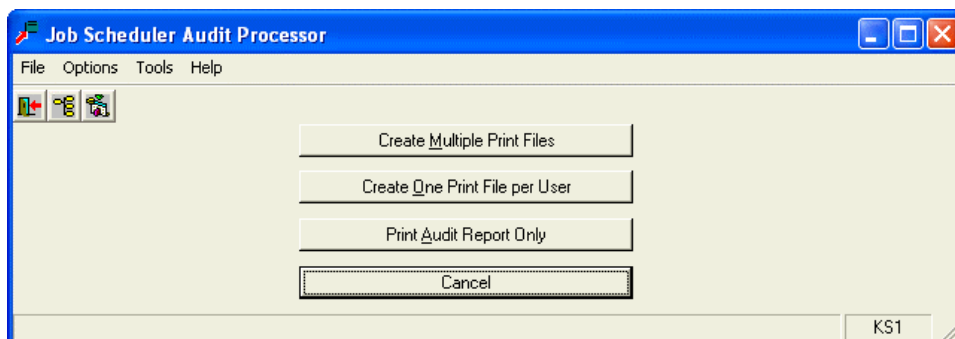
### *Introduction to the Audit Processor*

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
<b>Create Multiple Print Files</b>	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.
<b>Create One Print File per User</b>	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.
<b>Print Audit Report Only</b>	Create only the bsau report without creating any reports for users.
<b>Cancel</b>	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
-----
DISTRIB1 RPT1000B CIGL 05/21/07 12:16 12:16 F /ctron_3.1r/exec/spool/ccprt/rpt10001
PAGES
1      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

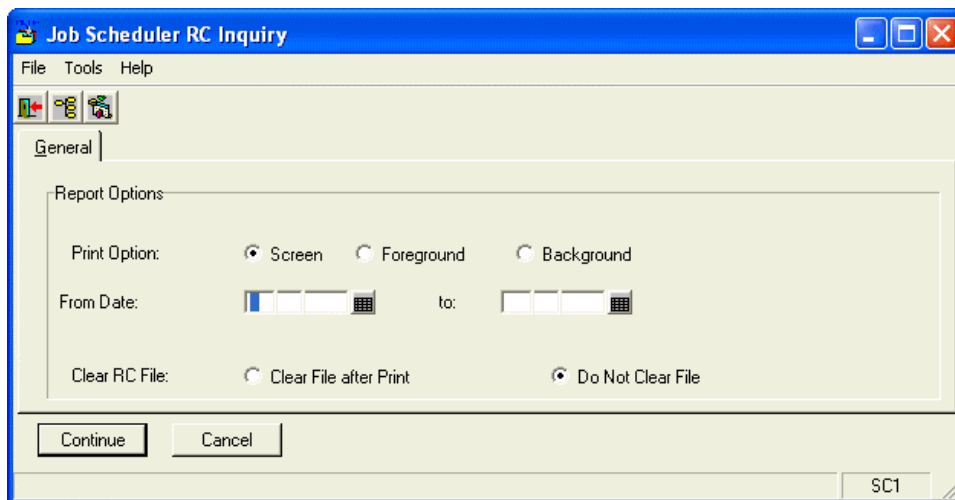
### *Introduction to Return Code Inquiry/List*

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.

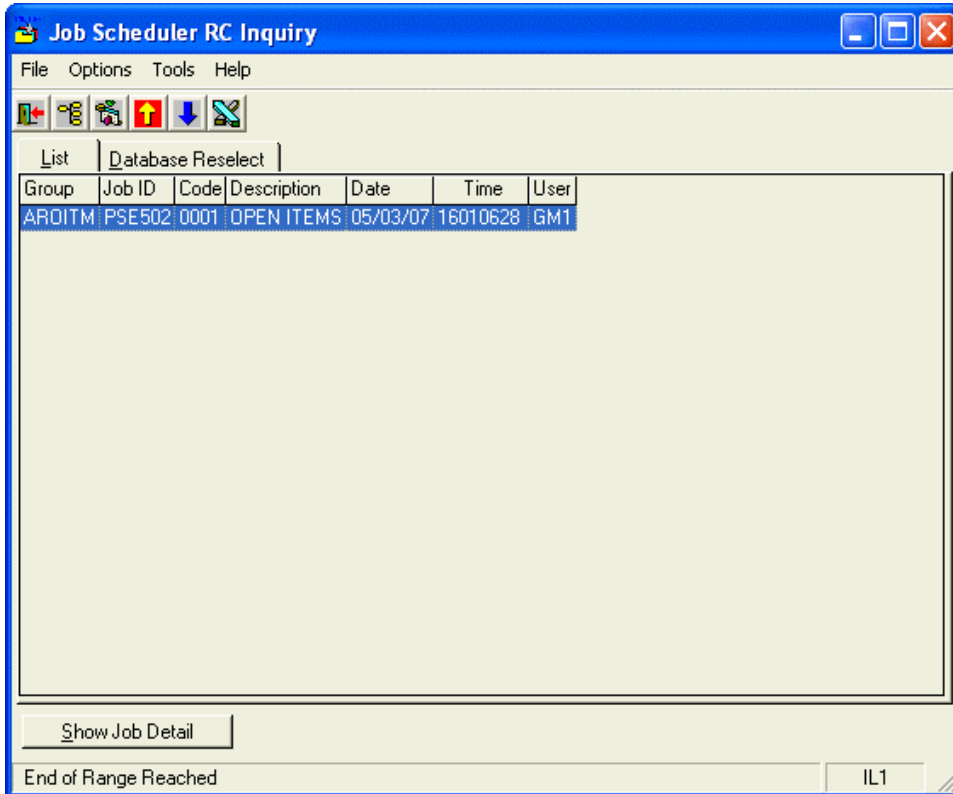
*Default:* Do Not Clear File

☞ *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 Screen Option

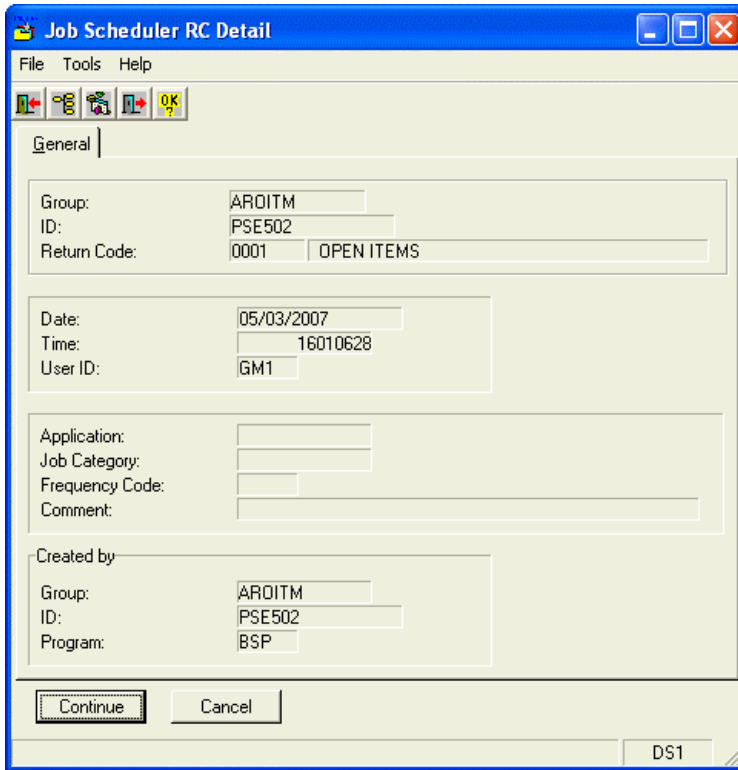
When the Screen option is selected, an inquiry window displays.



**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.

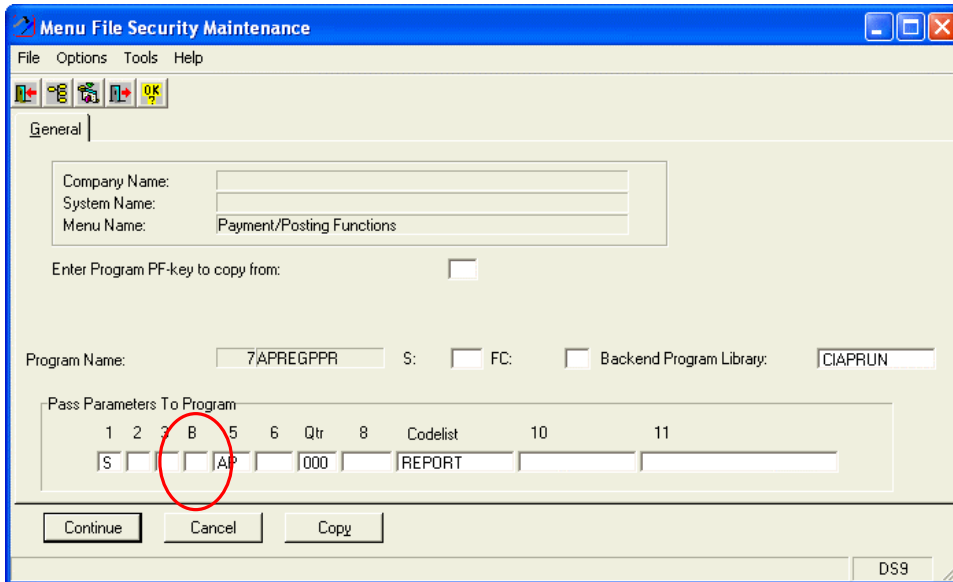
## Return Codes Print Option

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  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.
    
```





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).

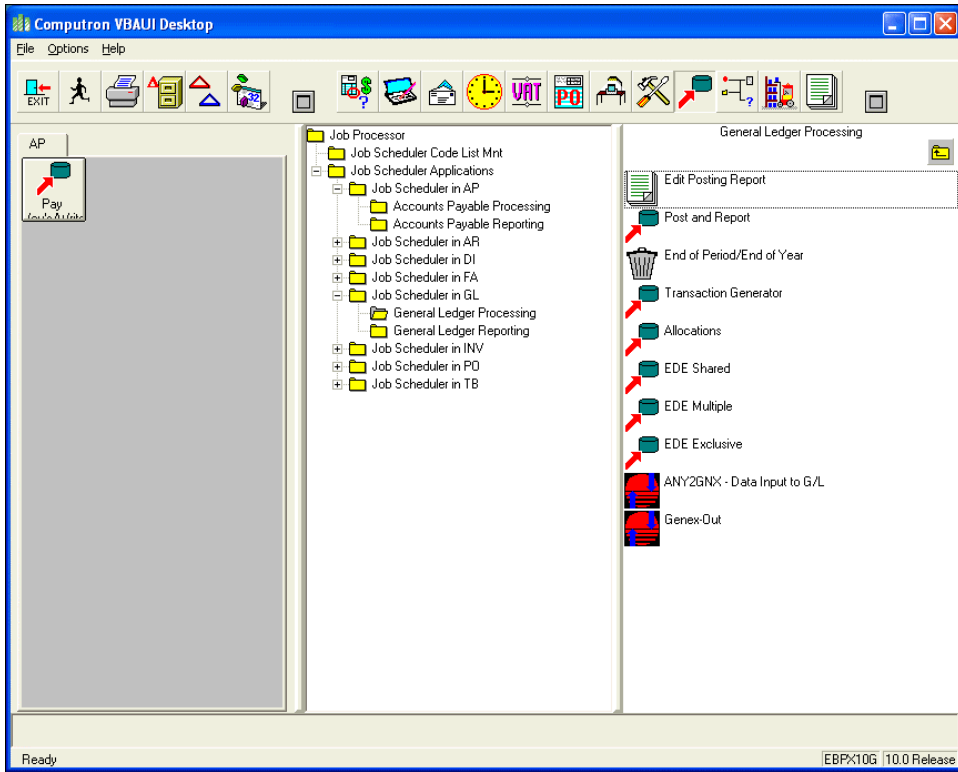
B – This value indicates that the Job Scheduler Maintenance Window (BS1) is to display. However, by pressing the Continue pushbutton on this window, the user can bypass this window and continue with the other windows in the function.

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.

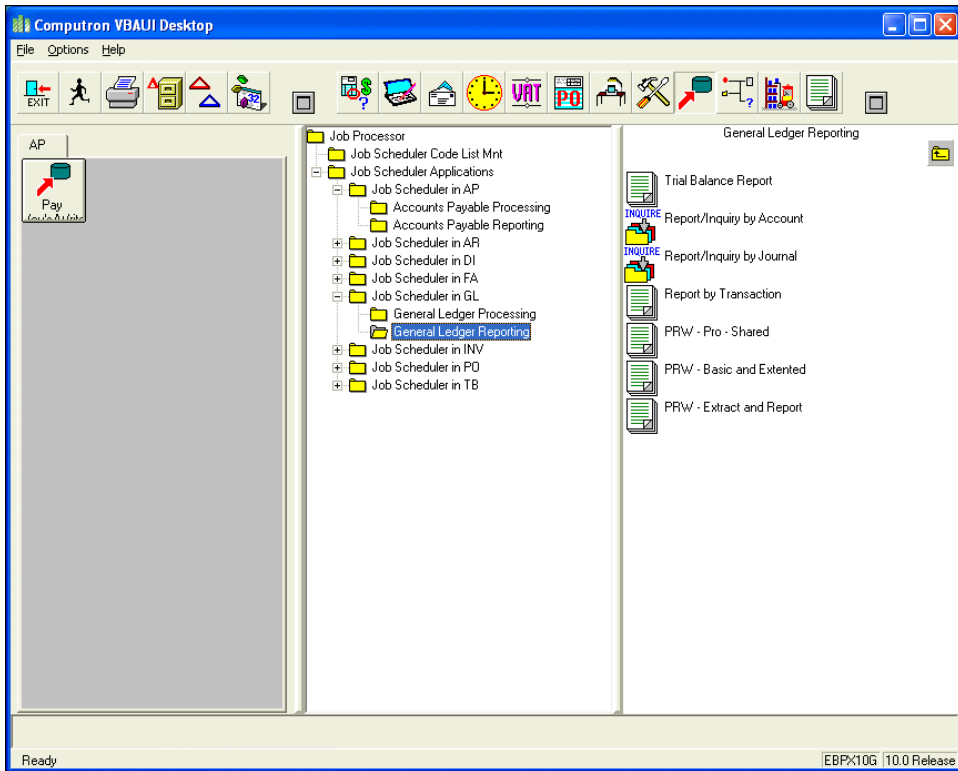
## Job Processor Only Menus

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.

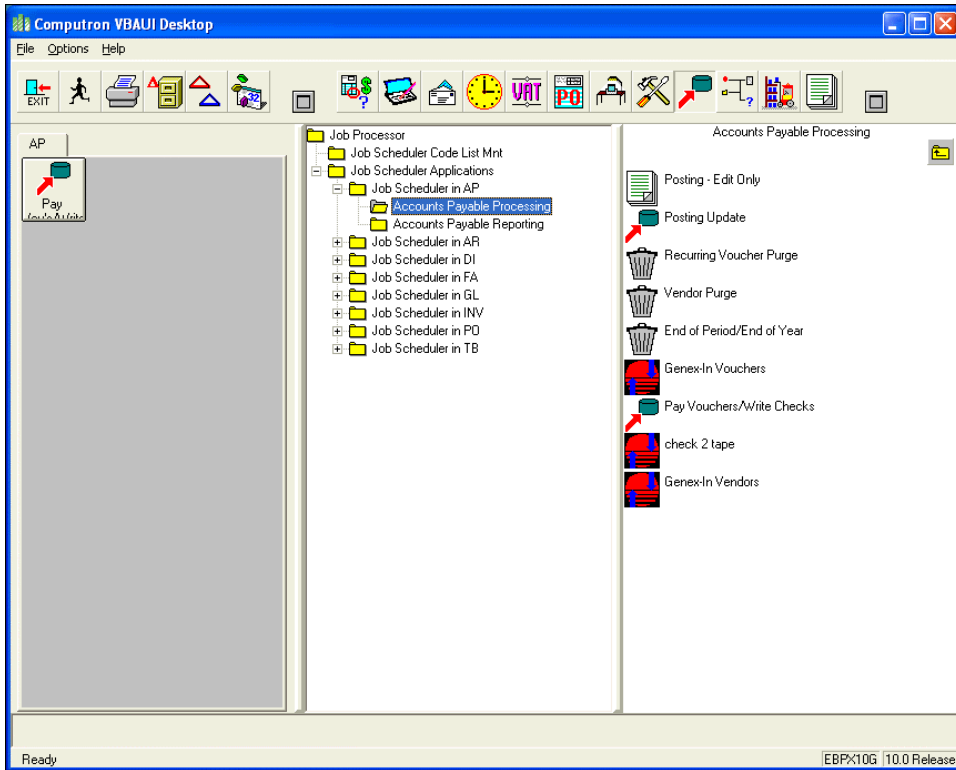
*Note:* These menus are released with the system, but can be modified or replaced completely at a client site. In addition, non- Computron functions can be placed on the menu and parameters established for processing using the same method as for Computron functions.



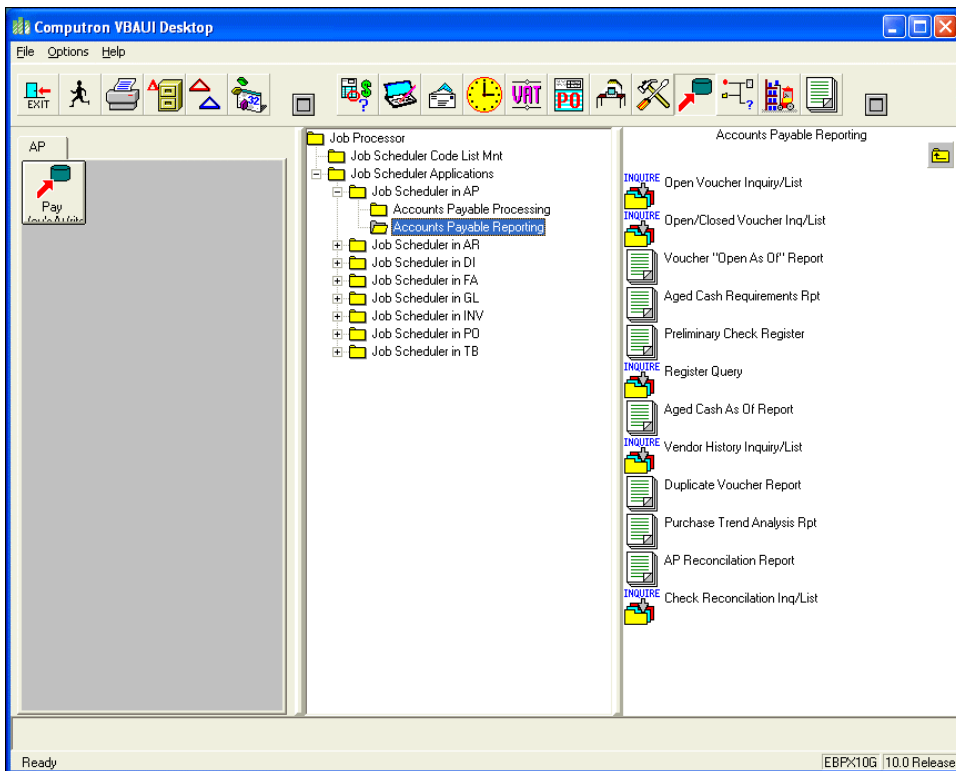
**GL Processing Functions**



**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/_ctron_` (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 Windows 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 Windows 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 Windows 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. The table of each possible return code in the Job Processor return code table is below.

## 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**.

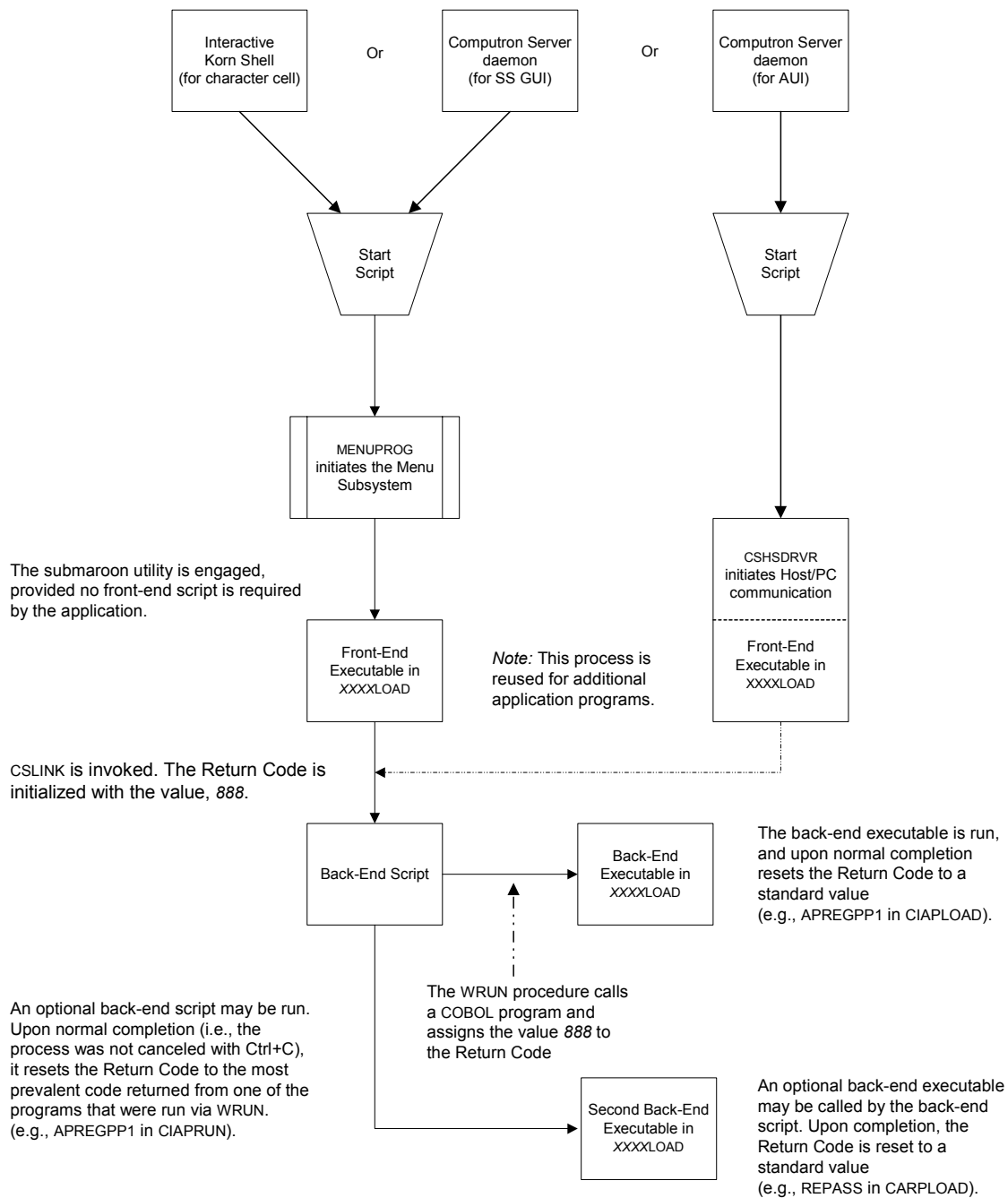
<b>Return Code</b>	<b>Description</b>
<b>888</b>	Process canceled by operator.
<b>889</b>	Process canceled by CSDISASTER.
<b>997</b>	Process canceled due to submaroon failure.
<b>999</b>	Link failed.
<b>2101</b>	Previous job was not run. Job Processor record not found. Job skipped (BSP only).
<b>8101</b>	Previous job was canceled in the Job Processor. Job skipped (JP only).
<b>1101</b>	Return Code not in range checked. Job skipped (JP only).
<b>000</b>	OK.
<b>105</b>	OK, but no records found.
<b>102</b>	OK, but warnings found.
<b>101</b>	Errors encountered.
<b>111</b>	Errors encountered evaluating the selection criteria or other values.

Return Code	Description
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).
126	Program or report was not found.
131	Program was interrupted by user.

## Illustrations

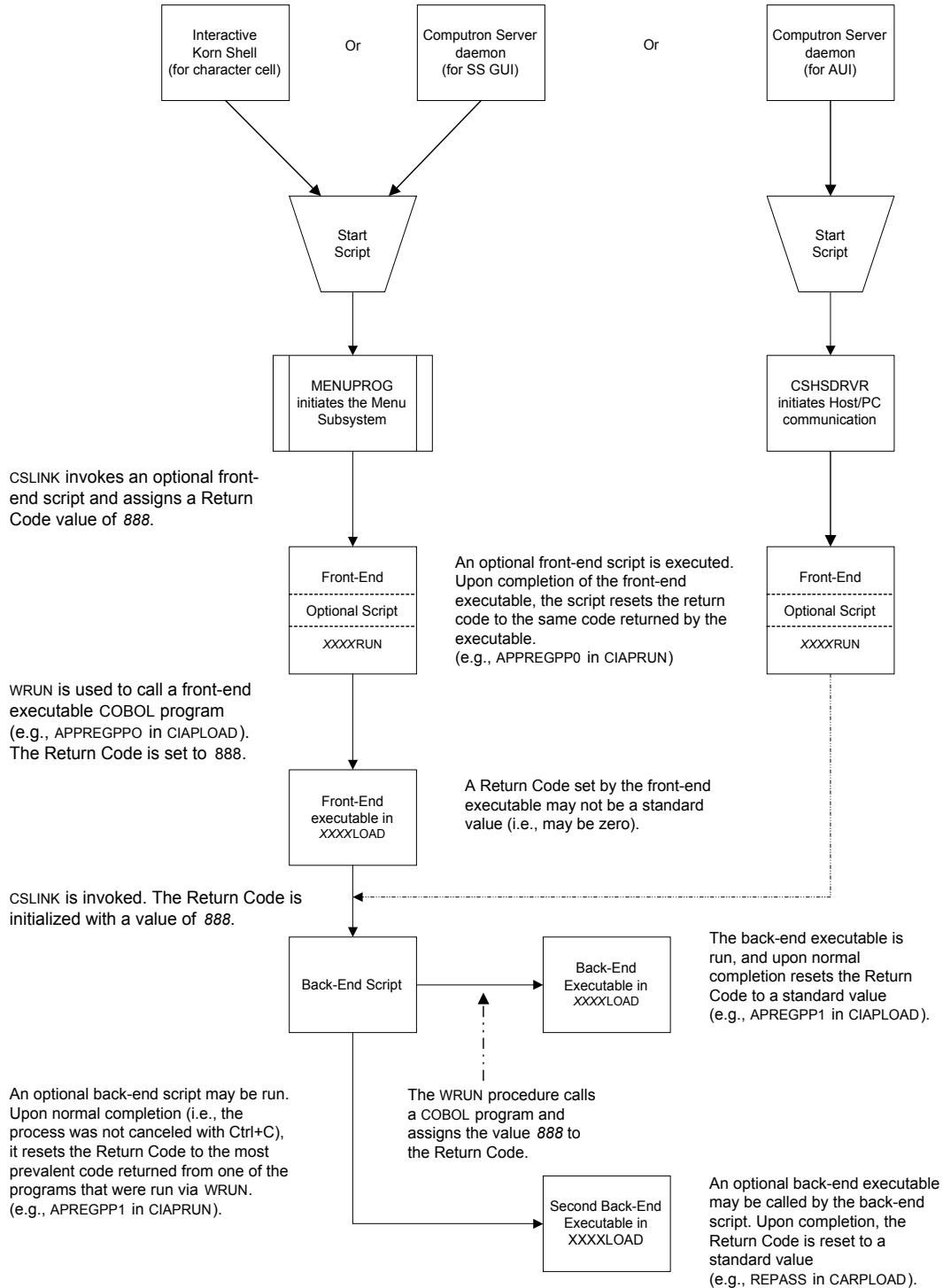
The flow chart below illustrates the flow of processes generated interactively, using the submaroon utility.

**Figure 1: Application Process Flow Using Submaroon**



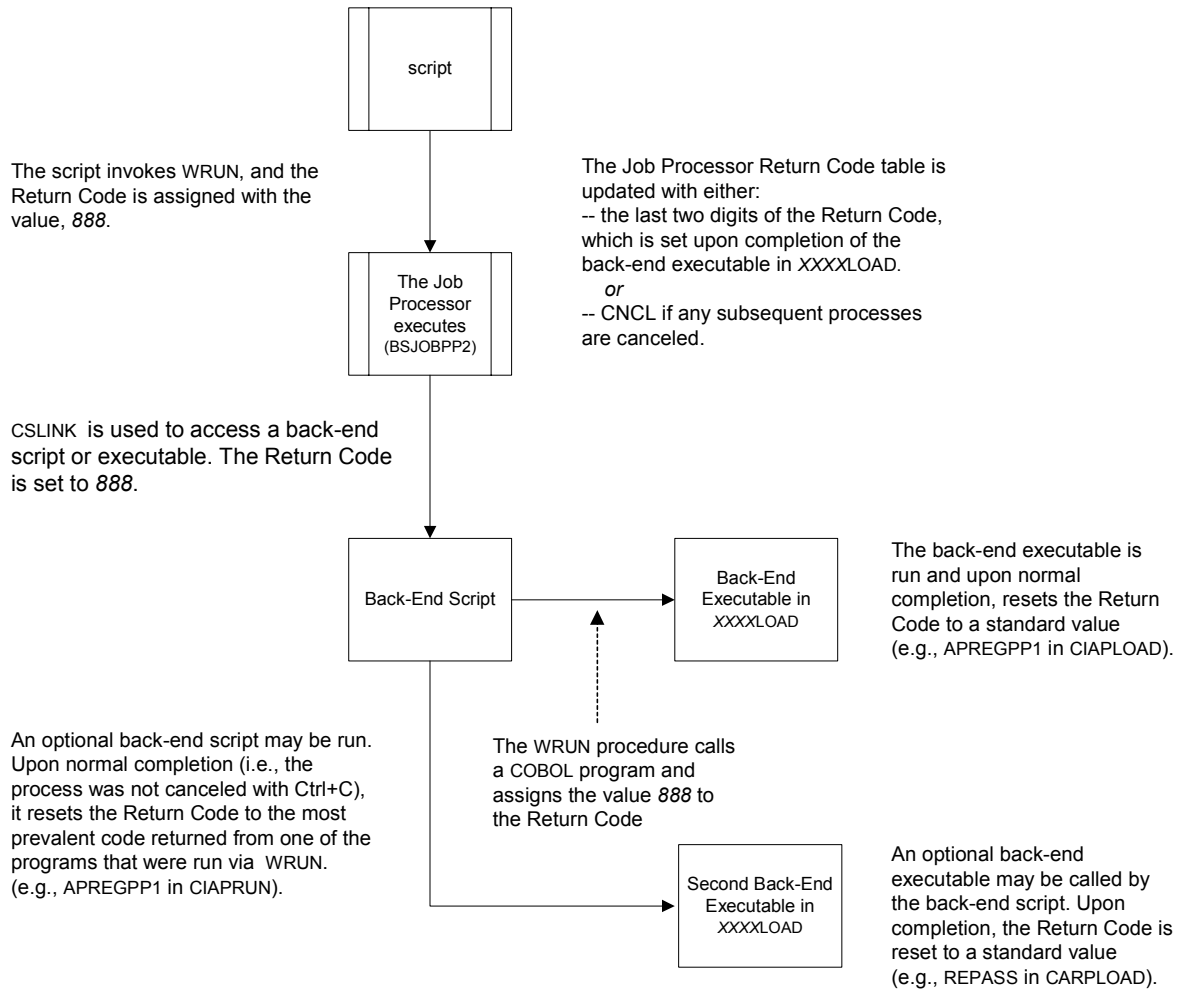
The figure below represents the flow of processes generated interactively, when the MENUPROG program invokes a script using CSLINK.

**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="20071101" 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*

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

In the Windows environment, this utility can only be accessed via the NTStart utility, which opens a command-line Windows console session. NTStart is a program shell that sets up the environment variables for a specific Computron environment. It then presents the user with a shell prompt “\$” so that they can run command line utilities in that environment.

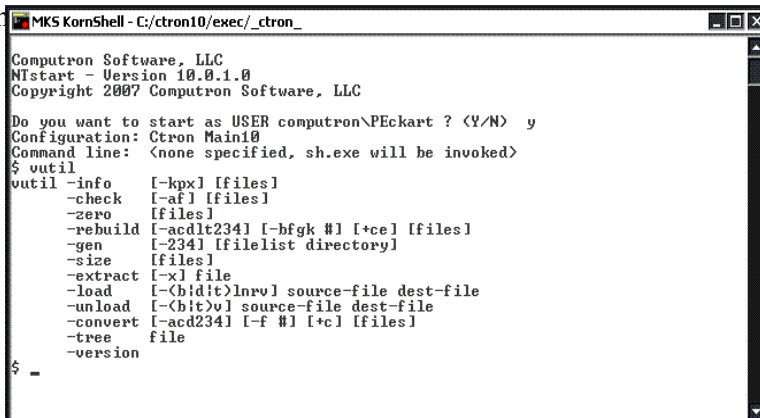
For example, run **ntstart –c“Ctron Main10”** to set up the environment for the “Ctron Main10” service. You are prompted for a user/password (must be valid in this environment). Once entered, you get a shell prompt (\$). From here, you can do things like create\_tables to manage a database, run scripts to do data conversions, etc. Since the NTStart utility was started in the “Ctron Main10” environment, all of the commands run within the shell session are executed on the files in the “Ctron Main10” environment. When done, type “exit” to go back to the standard Windows (DOS-like) command window.

### *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 via this utility. 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.

To run this program, go to the NTStart command prompt and issue a vutil command. The vutil command is detailed in detail later.



```

MKS KornShell - C:/ctron10/exec/_ctron_
Computron Software, LLC
NTstart - Version 10.0.1.0
Copyright 2007 Computron Software, LLC

Do you want to start as USER computron\PEckart ? (Y/N) y
Configuration: Ctron Main10
Command line: <none specified, sh.exe will be invoked>
$ 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 [-<bit>lnrw] source-file dest-file
      -unload [-<bit>v] source-file dest-file
      -convert [-acd234] [-f #] [+c] [files]
      -tree file
      -version
$ _
    
```

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.

- # # indicates the version of ACUCOBOL to use. –2 indicates 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.
- 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.
- 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.
- +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.
- 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.
- 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, 1 indicates the first alternate, etc.)
- m This option specifies the size of the spool media in bytes. –m is also used in conjunction 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.
- 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.

*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*

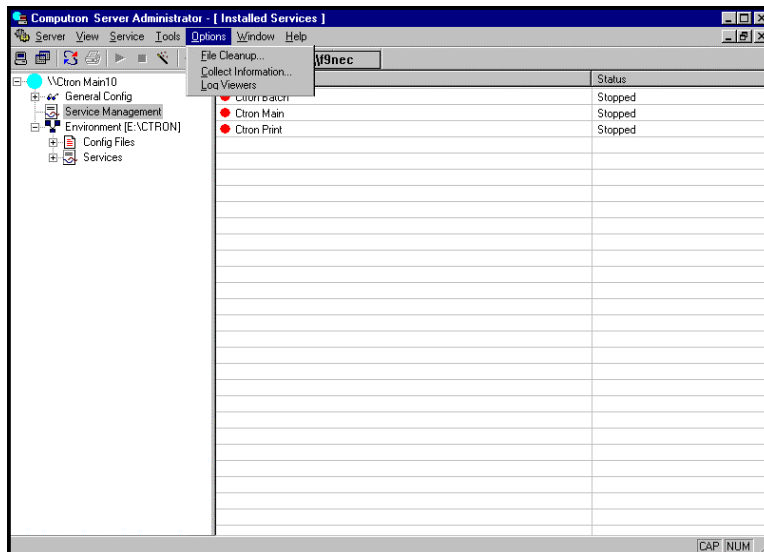
This function generates a listing of all programs and subroutines in a specified volume/library. The following example demonstrates how to generate a program listing of the General Ledger load library.

☞ *Note:* This function is available via the character cell presentation, the Computron Server Administrator or the NTStart utility. For character cell, access this function under the Universal Utilities menu, System Utilities Menu.

### *Running Program Report List using “Collect Information” in the Computron Server Administrator*

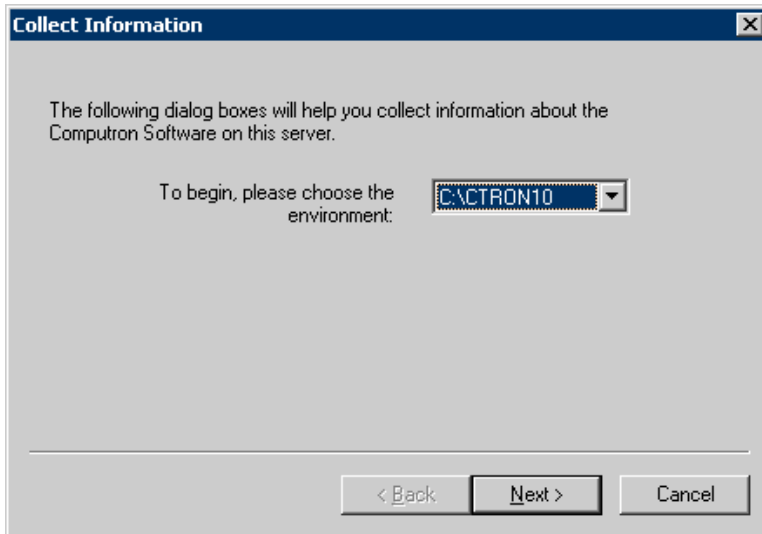
☞ *Note:* For more information on using “Collect Information,” refer to the Section entitled “Collecting Information using the Computron Server Administrator.”

1. Launch Computron’s Server Administrator.
2. Select the Options menu and the Collect Information option.



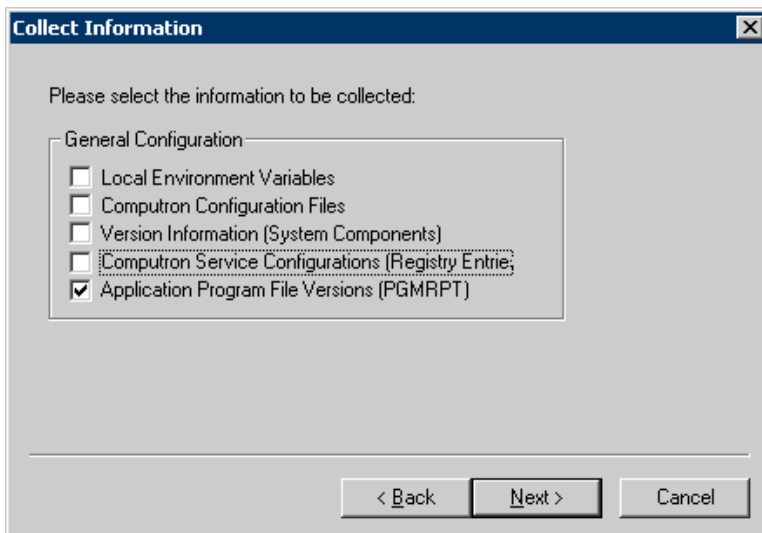
**Computron Server Administrator – Service Management**

3. Select the environment from where the Program Report list is needed.



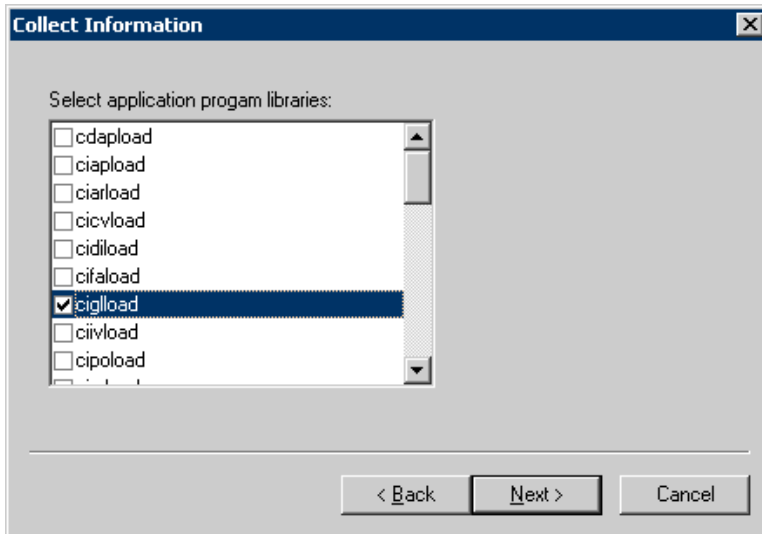
**Collect Information Window – Choose Environment**

4. Click Next.



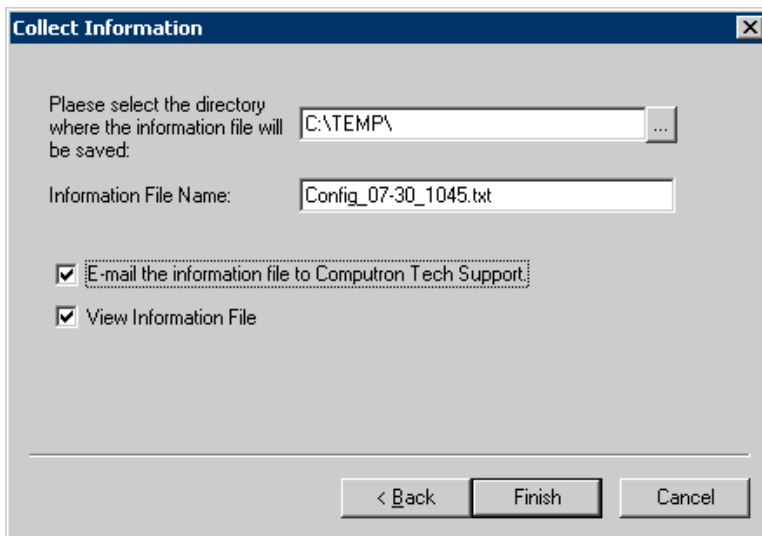
**Collect Information Window – Information to be Collected**

5. Check the Application Program File Versions (PGMRPT) field. Leave all other fields blank.
6. Click next.



**Collect Information – Program Libraries**

7. Select each of the desired program libraries.
8. Click Next.



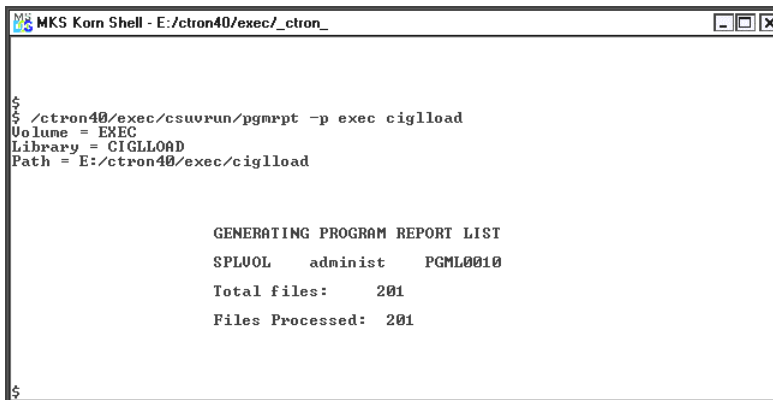
**Collect Information – Define Output for Information Collection**

9. Select the directory to place the “Program Report List” output in. Enter a name for the output file that is created. Select the E-mail the information file to Computron Tech Support field if you want the file to be emailed. Select the View Information File field if you want to display the report after it is generated.
  - ☞ *Note:* In order to automatically email the “Program Report list,” it is necessary to have a MAPI compliant email client (e.g., Outlook) installed on the machine where the Program Report List is being run.
10. Click Finish.

## Running Program Report List using NTStart

In the Windows environment, this utility can only be accessed via the NTStart utility, which opens a command-line console session.

1. Initiate an NTStart session.
2. If the MKS Korn Shell is not automatically loaded, load it now by issuing the SH.EXE command.
3. Enter the following command line:  
n:/.../exec/csvrun/pgmrpt -p exec cigload



```
MKS Korn Shell - E:/ctron40/exec/_ctron_
$
$/ctron40/exec/csvrun/pgmrpt -p exec cigload
Volume = EXEC
Library = CIGLOAD
Path = E:/ctron40/exec/cigload

          GENERATING PROGRAM REPORT LIST
          SPLUOL  administ  PGML0010
          Total files:    201
          Files Processed: 201
$
```

Program Report List Window

☞ *Note:* The -p parameter copies the report listing into your print queue. To view the listing online, simply remove the -p parameter from the command line.

### Volume

Enter the volume to interrogate.

### Library

Enter the library to interrogate.

4. Exit the NTStart session.

```

Telnet supsun2
File PGML0003 PSEPRT SPLUOL IP1DIS
< 1>Menu < 3>Record < 5>Next < 7>Up < 8>Find
< 9>Left <10>Right <14>Pos <15>Print <16>E&xit

May 3 14:09 2007 PROGRAM REPORT LIST FOR /enterprise/10qa/exec/ciglload Page 1

COCOMPI 10.00.00 SBEISOL 10.00.00
CODIMIDS 10.00.00 SBEISSUC 10.00.00
COGLCOMP 10.00.00 SBENCRIN 10.00.00
COLEDGER 10.00.00 SBENTIDS 10.00.00
COPAI 10.00.00 SBENTITY 10.00.00
COPOSTAC 10.00.00 SBEXCLIC 10.00.00
COSCNRP 10.00.00 SBEXTSEL 10.00.00
COTRANS D 10.00.00 SBFMT4PC 10.00.00
COTRANSH 10.00.00 SBFMTXT 10.00.00
COTRANGNX 10.00.00 SBFMTOLN 10.00.00
COWEBTB 10.00.00 SBFUND D 10.00.00
COWEBTBX 10.00.00 SBGL2INQ 10.00.00
CURRUPLD 10.00.00 SBGLENV 10.00.00
GLOBUMNT 10.00.00 SBGLRELS 10.00.00
GLTAREX2 10.00.00 SBGNXCNU 10.00.00
  
```

Online Inquiry — Program Report List Output

## Program Report Listing

### Sample Computron Program Report List

Jul 30 11:24 2007 PROGRAM REPORT LIST FOR C:/ctron10/exec/ciglload Page 1

COCOMPI	10.00.00	SBEISAUD	10.00.00
CODIMIDS	10.00.00	SBEISIDS	10.00.00
COGLCOMP	10.00.00	SBEISOL	10.00.00
COLEDGER	10.00.00	SBEISSVC	10.00.00
COPAI	10.00.00	SBENCRIN	10.00.00
COPOSTAC	10.00.00	SBENTIDS	10.00.00
COSCNRP	10.00.00	SBENTITY	10.00.00
COTRANS D	10.00.00	SBEXCLIC	10.00.00
COTRANSH	10.00.00	SBEXTSEL	10.00.00
COTRANGNX	10.00.00	SBFMT4PC	10.00.00
COWEBTB	10.00.00	SBFMTXT	10.00.00
COWEBTBX	10.00.00	SBFMTOLN	10.00.00
CURRUPLD	10.00.00	SBFUND D	10.00.00
F2000GP	09.00.00	SBGL2INQ	10.00.00
GLOBVMNT	10.00.00	SBGLENV	10.00.00
GLTAREX2	10.00.00	SBGLRELS	10.00.00
GLTAREXIT	10.00.00	SBGNXCNU	10.00.00
GLUEXIT		SBGSTRNG	10.00.00
GLWFEXIT	09.00.00	SBID2CL	10.00.00
JPGLEXIT	10.00.00	SBIDMNT	10.00.00
PAIEX2	10.00.00	SBIVRGN	10.00.00
PAIEXIT	10.00.00	SBLDGNEW	10.00.00
PAIVAL	10.00.00	SBLINPRT	10.00.00
S2CHART	10.00.00	SBOLAPGL	10.00.00
S2ENTIT	10.00.00	SBOLPSTR	10.00.00
S2PCENT	10.00.00	SBPAIBAL	10.00.00
S2SGPAT	10.00.00	SBPCPORT	10.00.00
S2SRCEJ	10.00.00	SBPCWAIT	10.00.00
SBALEAGL	10.00.00	SBPIDEF	10.00.00
SBAPCAL	10.00.00	SBPIDLNK	10.00.00
SBARCAL	10.00.00	SBPIPORT	10.00.00
SBAVLDEL	10.00.00	SBPISTR	10.00.00
SBBCMLIC	10.00.00	SBPISVC	10.00.00
SBBDGEN	10.00.00	SBPIWAIT	10.00.00
SBBDWIO	10.00.00	SBPORG	10.00.00
SBBDPAI	10.00.00	SBPRTPAS	10.00.00
SBBDPAR	10.00.00	SBRELMT	10.00.00
SBBDRT	10.00.00	SBRLDRVR	10.00.00
SBBDUSVC	10.00.00	SBRPTPRT	10.00.00
SBBDUP	10.00.00	SBSLMNT	10.00.00
SBCOLPRT	10.00.00	SBSLPRT	10.00.00
SBDACK	10.00.00	SBTEDIT	10.00.00
SBDCLBAL	10.00.00	SBTFRLIC	10.00.00
SBDLLDG	10.00.00	SBTRANRV	10.00.00
SBDNSEL	10.00.00	SBTRINQ	10.00.00
SBUDALT	10.00.00	calchng	10.00.00
SBUPSHDW	10.00.00	capisvc	10.00.00
SBWFDEL	10.00.00	caxlsvc	10.00.00
SBXATTR	10.00.00	cmpldg1	10.00.00
SBXLADHC	10.00.00	cnvledgr	10.00.00
SBXLCLS	10.00.00	coldefmt	10.00.00
SBXLSVC	10.00.00	colmnt	10.00.00
SBXLTMPL	10.00.00	comboardd	10.00.00
SBXLUPLD	10.00.00	convacc	10.00.00
abalmnt	10.00.00	copsmtt	10.00.00
accdefmt	10.00.00	crmstrd0	10.00.00
accmnt	10.00.00	crmstrdl	10.00.00
acct110	10.00.00	currgrnx0	10.00.00

acctill	10.00.00	currngx1	10.00.00
acctmnt	10.00.00	currmnt	10.00.00
acctrgn0	10.00.00	defprt0	10.00.00
acctrgn1	10.00.00	defprt1	10.00.00
adbcomp0	10.00.00	dellldg0	10.00.00
adbcomp1	10.00.00	dellldg1	10.00.00
allocgn0	10.00.00	dmdtmnt	10.00.00
allocgn1	10.00.01	dnlahx10	10.00.00
allocmnt	10.00.02	dnlbud0	10.00.00
any2gnx0	10.00.00	dnlbud1	10.00.00
any2gnx1	10.00.00	dnlbudia	10.00.01
aparcsch0	10.00.00	dnlbudib	10.00.01
aparcsch1	10.00.00	dnlbtmpl	10.00.00
attrrgn0	10.00.00	drillbk0	10.00.00
attrrgn1	10.00.00	dsplyprt	10.00.00
ac.adb	Unable to get ve	edespmnt	10.00.00
afiedt.buf	Unable to get ve	eidsaud	10.00.00
auditpr0	10.00.01	eisext0	10.00.00
auditpr1	10.00.00	eisext1	10.00.00
avlfix0	10.00.00	encyend0	10.00.00
avlfix1	10.00.00	encyend1	10.00.00
avlfndel	10.00.00	envnmnt	10.00.00
balentry	10.00.00	eoepoy0	10.00.00
bentry	10.00.00	eoepoy1	10.00.00
bldaltp0	10.00.00	extdnl0	10.00.00
bldaltpl	10.00.00	extdnl1	10.00.00
bldshdw0	10.00.00	extrpg2i	10.00.00
bldshdw1	10.00.00	extrpgn0	10.00.00
budupld0	10.00.00	extrpgn1	10.00.00
budupwf	10.00.00	extrpgn2	10.00.00
caeissvc	10.00.00	f2000b1	10.00.00
calchn0	10.00.00	f2000b2	10.00.00
f2000b3	10.00.00	postrgn0	10.00.01
f2000f	10.00.00	postrgn1	10.00.00
fixldgr	10.00.00	pgtrsv1	10.00.00
fixmni	10.00.00	pstdbgn0	10.00.00
fixpadt1	10.00.00	pstdbgn1	10.00.00
fixperl3	10.00.00	pstdtrn0	10.00.01
fixpstac	10.00.00	pstdtrn1	10.00.01
fixrebal	10.00.00	relmnt	10.00.00
fxldgtx	10.00.00	repdefmt	10.00.00
genexgen	10.00.04	repgena	10.00.00
genexrst	10.00.00	repgenb	10.00.00
gldbopn	10.00.00	repgenic	10.00.00
gltrmf0	10.00.00	repsch0	10.00.00
gltrmf1	10.00.00	repsch1	10.00.00
glviutst	10.00.00	repsch2i	10.00.00
glvreset	10.00.00	rerungnx	10.00.00
gnxifmnt	10.00.00	rpt1000b	10.00.01
gtprtpas	10.00.00	rpt1000f	10.00.03
ledgjn10	10.00.03	rptdmnt	10.00.00
ledgjn11	10.00.00	rptgenb	10.00.00
ledgnab0	10.00.00	schmnt	10.00.00
ledgnab1	10.00.00	segmnt	10.00.00
ledgnw6	10.00.06	slmnt	10.00.00
ledgnw6_orig	10.00.05	stddmnt	10.00.00
ledgnw7	10.00.02	tedit	10.00.00
lindefmt	10.00.00	tentry	10.00.00
linemnt	10.00.00	testtr	10.00.00
linesil0	10.00.00	trangen0	10.00.00
linesil1	10.00.00	trangen1	10.00.00
linverf0	10.00.00	tranrev0	10.00.00
linverf1	10.00.00	tranrev1	10.00.00
massbud	10.00.00	transinq	10.00.00
mstrdel1	10.00.00	transpp0	10.00.00
pcpstil0	10.00.00	transpp1	10.00.01
pidback	10.00.00	traumnt	10.00.00
portinit	10.00.00	trevinq	10.00.09
postcop0	10.00.01	trevinq.10.02	10.00.02
postcop1	10.00.01	trevrpt0	10.00.08
postedel	10.00.00	trevrpt0.10.02	10.00.02
postedfx	10.00.00	trevrpt1	10.00.07
postedit	10.00.00	trevrpt1.10.02	10.00.02
postfix	10.00.00	trmsfex0	10.00.00
postil0	10.00.03	trmsfex1	10.00.00
postill	10.00.00	trmsfin0	10.00.00
postmnt	10.00.00	trmsfin1	10.00.00
		trmslimt	10.00.00
		trmslpmt	10.00.00
		upldede	10.00.00

## 4.4 Service Log File

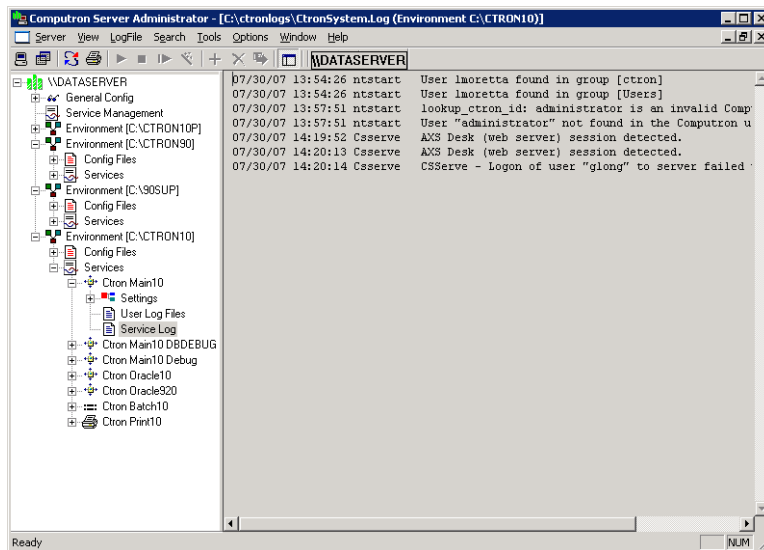
### Introduction to the Service Log File

This function is used to display the contents of the Windows Service Log file. It contains a running list of informational messages that occur during a session.

*Note:* This function is available via the Computron Server Administrator. For more information about Computron Server Administrator, refer to Section 5.1, “Computron Server Administrator.”

### Log File Window

The following window displays when you select the Service Log from the Service Log option from the Services menu:



**Computron Service Log File Window**



## 4.5 Clear Print Pass File

### *Introduction to the Clear Print Pass File Function*

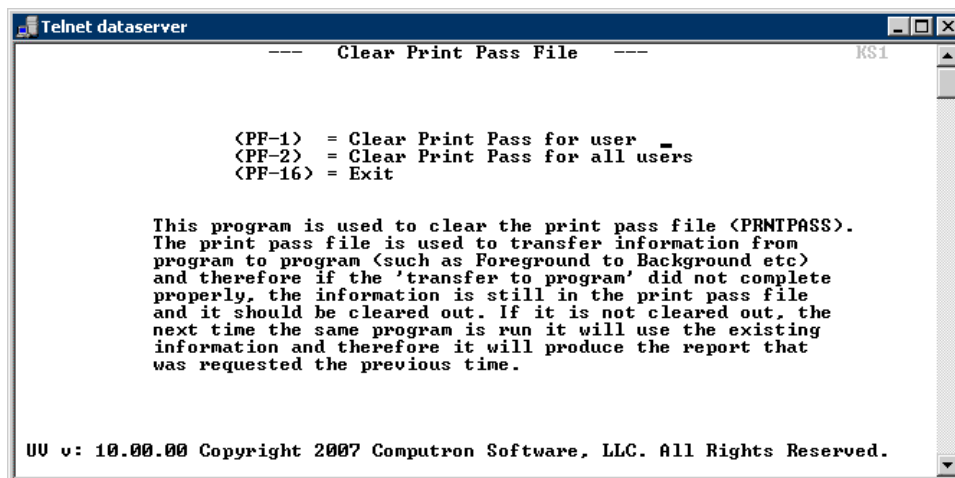
This function is used to clear Print Pass file (PRNTPAS4) records/entries created by various Computron functions. A Print Pass file is created for each database at the time of installation. This file contains one record for each function run in front-end/back-end mode; i.e., any function that allows 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 dataserver
--- 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.

<b>Options for the Clear Print Pass File Window</b>		
<b>Function Key</b>	<b>Name</b>	<b>Description</b>
<b>PF-1</b>	Clear Print Pass for user	Clears the print pass file records created for a specified user's ID.
<b>PF-2</b>	Clear Print Pass for all users	Clears the print pass file of all records, regardless of user ID.
<b>PF-16</b>	Exit	Returns to the Universal Utilities menu without making changes to the Print Pass file.

## 4.6 Clear Hold List Entries

### *Introduction to Clear Hold List Entries*

The Clear Hold List Entries function (CSHLDCLR) removes held resources associated with 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:* There are 2 ways to clear a hold list entry. By using “Computron’s Process Monitor” or using the “cshldclr” utility. Computron recommends using “Computron’s Process Monitor”

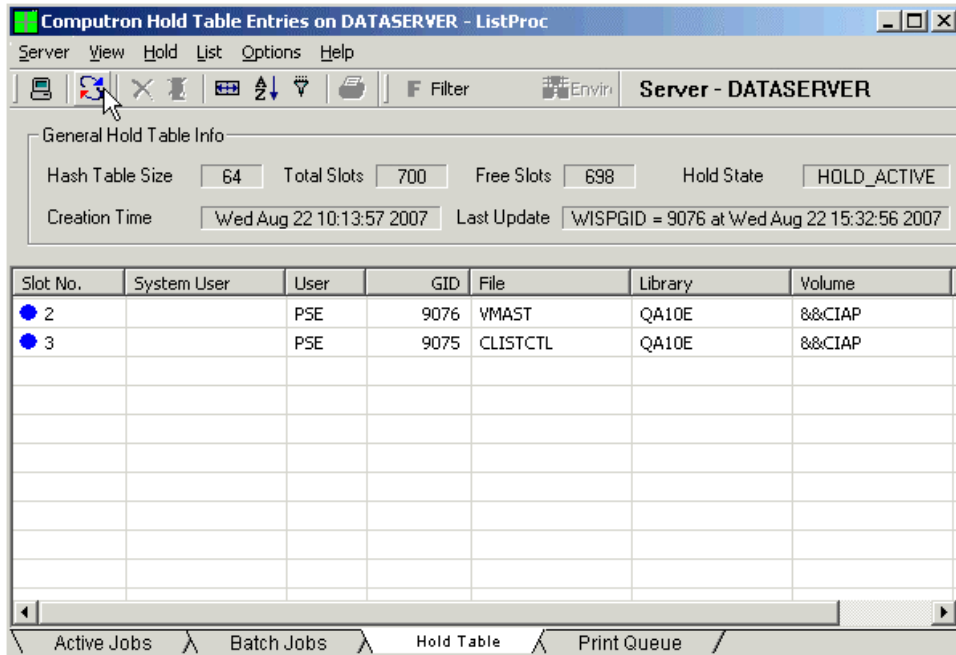
⚠ *Warning!* When clearing one entry from the hold list, all of those user’s entries in the hold list are removed. Therefore, before clearing a hold list entry, verify that the user with the invalid hold list entry is logged off!

### Managing the Hold List using Computron’s Process Monitor

#### *Deleting a Hold List entry*

☞ *Note:* For additional information on Computron’s Process Monitor, see Section 5.11, “Computron’s Process Monitor.”

1. Launch Computron’s Process Monitor.
2. Click on the Hold Table tab.



**Process Monitor Hold Table Tab**

3. Highlight the Hold list entry by clicking on it.

4. Click the delete button .

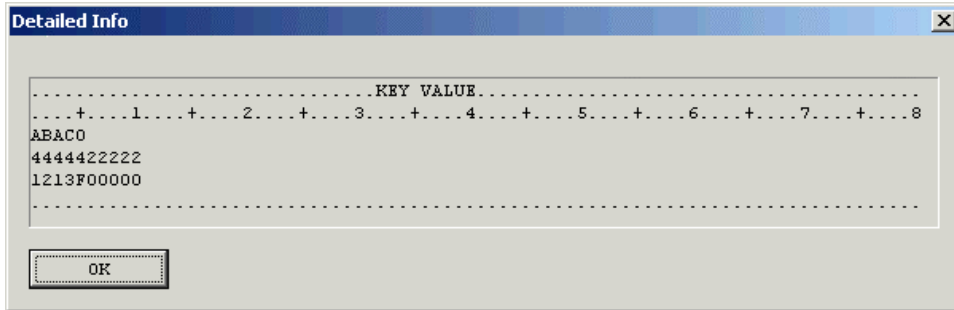
5. Click Yes to confirm clearing of the hold list for that user.

**Clear User's hold list confirmation**

*View Key Details*

Key Details shows the primary key of item that is being held.

1. Launch Computron's Process Monitor.
2. Click on the Hold Table tab.
3. Click on a Hold item.
4. On the Hold menu, select Entry and click Key Details on the submenu.

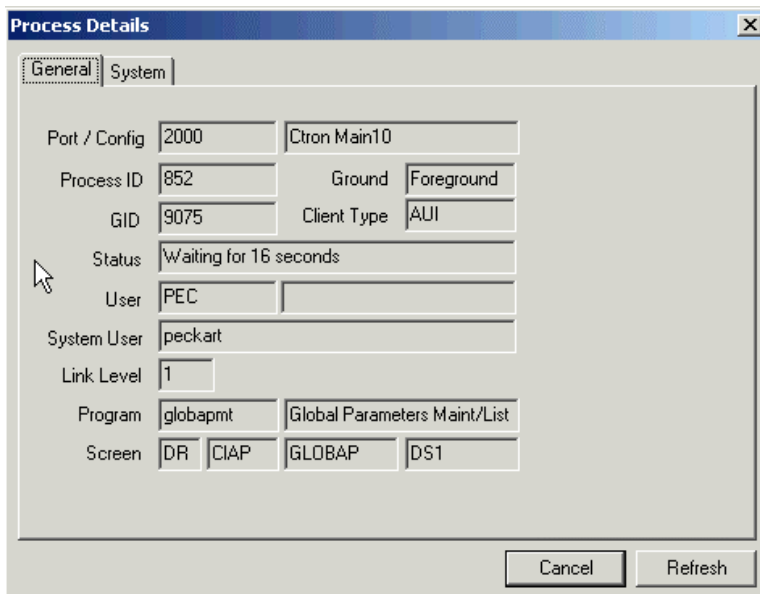


Detailed Info dialog box

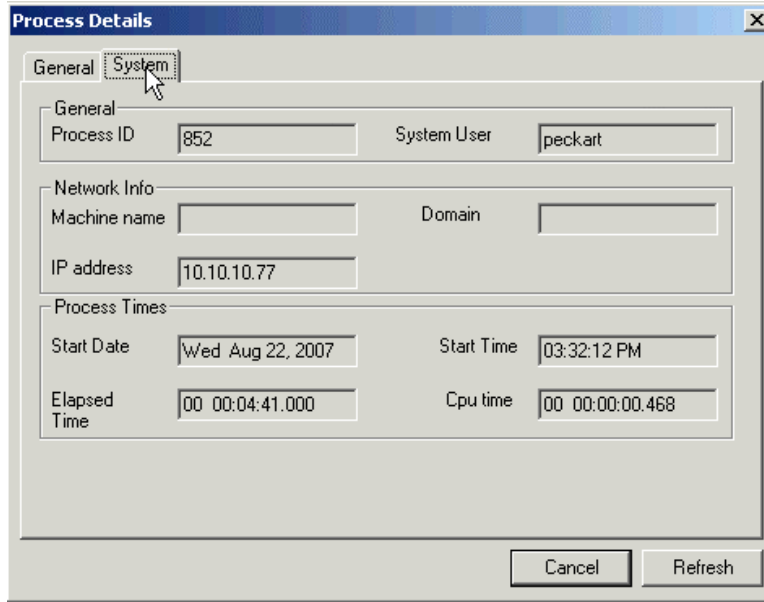
### Viewing the Hold Process Details

Process Details describe the process that originally created the hold request.

1. Launch Computron's Process Monitor.
2. Click on the Hold Table tab.
3. Click on a Hold item.
4. On the Hold menu, select Entry and click Hold Process Details on the submenu.



Process Details – General Tab dialog box



Process Details – System dialog box

## Running the Clear Hold List Function using NTStart

*Note:* In the Windows environment, this utility can only be accessed via the NTStart utility, which opens a command-line NT console session.

1. Initiate an NTStart session.
2. If the MKS Korn Shell is not automatically loaded, load it now by issuing the `sh.exe` command.
3. Enter the following command line:

```
n:/.../exec/csuvrun/cshldclr
```

The following prompts are displayed:

```

cshldclr
Hold slots: 700, Hash table size: 64, Initialized: Wed Aug 22 10:13:57 2007
Last update: UID = 0, WISPGID = 9075 at Wed Aug 22 15:36:36 2007
cshold_state = HOLD_ACTIVE, cshold_count = 17, free_list_head = 1

HASH TABLE ENTRIES:
=====
0 -> 30 free, Hash entry 31 = 3, 32 -> 36 free, Hash entry 37 = 2,
38 thru 63 free

FREE HOLD LIST CHAINS:
=====
--> 1 --> 0 --> 4..699
Number of slots free: 698

HOLD RECORD ENTRIES:
=====
Hash table entry 31:
Printing hold slot # 3
WISPGID=9075, UID=0, Holder_Id=PSE
FILE=CLISTCTL LIB=QA10E UOL=&&CIAP <dev:5 ino:1632724869>
keylen=8
***End of chain***
-----
Hash table entry 37:
Printing hold slot # 2
WISPGID=9076, UID=0, Holder_Id=PSE
FILE=UMAST LIB=QA10E UOL=&&CIAP <dev:5 ino:13243102>
keylen=10
***End of chain***
-----

USER HOLD RECORD ENTRIES:
=====
User # holds User # holds User # holds User # holds
=====
PSE 2

Total number of hold entries used: 2

*** 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?

#### 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)?"
```

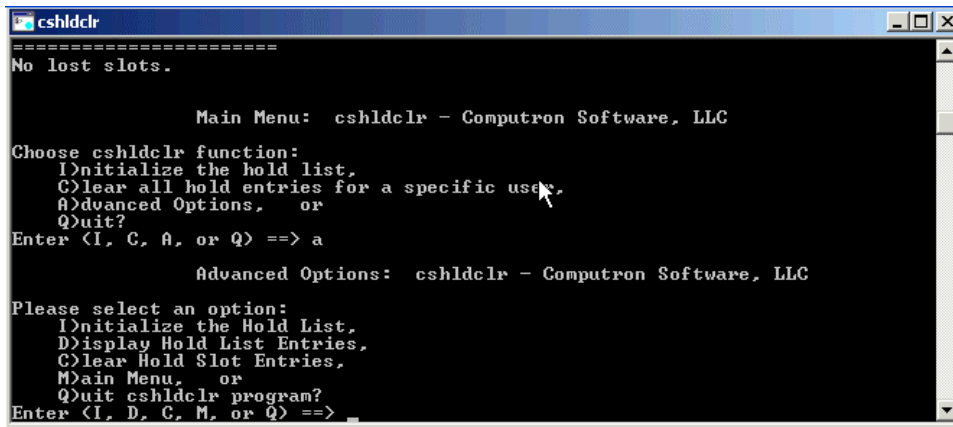
If all the holds are justified for clearing, enter Y to clear all holds for the listed IDs. Otherwise, enter N to return to the System Utilities menu.

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:



```

cshldclr
=====
No lost slots.

Main Menu: cshldclr - Computron Software, LLC

Choose cshldclr function:
I>initialize the hold list,
C>clear all hold entries for a specific user,
A>advanced Options, or
Q>quit?
Enter (I, C, A, or Q) ==> a

Advanced Options: cshldclr - Computron Software, LLC

Please select an option:
I>initialize 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.

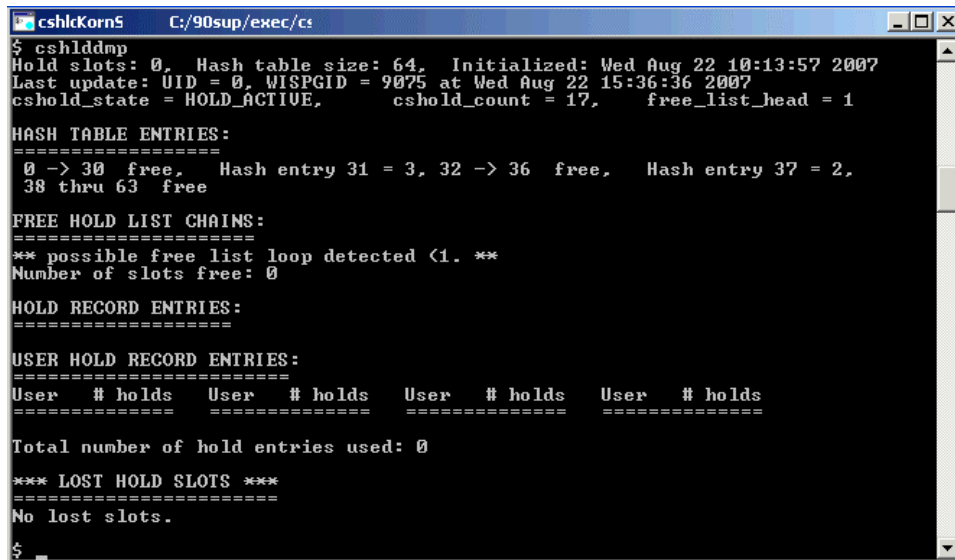


## Viewing Active Process IDs

The CSHLDDMP function 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.

1. Initiate an NTStart session.
2. If the MKS Korn Shell is not automatically loaded, load it now by issuing the sh.exe command.
3. Enter the following command line:

```
n:/.../exec/csuvrn/cshlddmp
```



```

cshlcKorn5 C:/90sup/exec/cs
$ cshlddmp
Hold slots: 0, Hash table size: 64, Initialized: Wed Aug 22 10:13:57 2007
Last update: UID = 0, WISPGID = 9075 at Wed Aug 22 15:36:36 2007
cshold_state = HOLD_ACTIVE, cshold_count = 17, free_list_head = 1

HASH TABLE ENTRIES:
=====
0 -> 30 free, Hash entry 31 = 3, 32 -> 36 free, Hash entry 37 = 2,
38 thru 63 free

FREE HOLD LIST CHAINS:
=====
** possible free list loop detected (1. **
Number of slots free: 0

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.

$
    
```

CSHLDDMP listing

## 4.7 Running Scripts on Windows

### *Introduction to Running Scripts on Windows*

This section explains how to run a Computron script from outside the Computron environment. For detailed instructions on creating internal Computron script procedures, refer to Section 5.5, “Creating Scripts using Computron’s Procedure Generator,” later in this guide.

- ◆ A program (e.g., Visual Basic) runs the NTStart utility (NTSTART.EXE);
- ◆ NTSTART.EXE sets up the Computron environment;
- ◆ NTSTART.EXE runs the Korn shell (SH.EXE);
- ◆ SH.EXE runs the Computron script;
- ◆ The Computron script runs the application.

The format of the command line is as follows (in a shorthand BNF).

```

commandline ::= nstartpath ntstart_parms

nstartpath  ::= full path of ntstart.exe (e.g.,
n:\...\exec\_ctron\_ntstart.exe

ntstart_parms ::= [ -c configuration_spec ] shell_command

configuration_spec ::= quote_literal config_name quote_literal
config_name = Ctron configuration_name

shell_command ::= quote_literal shellpath shell_parms
quote_literal

shellpath ::= full path of sh.exe, such as c:\mksnt\sh.exe

shell_parms ::= scriptpath

scriptpath ::= full path of script to run (e.g.,
n:/.../exec/ciglrn/myscript)

quote_literal ::= "
    
```

☞ *Note:* Forward slashes are used in the scriptpath variable.

### *Sample Script*

☞ *Important!* This script path is word-wrapped — therefore, you must use a single space where the string wraps to the next line:

```

n:\...\exec\_ctron\_ntstart.exe -c "Ctron Prod"
"c:\mksnt\sh.exe n:\...\exec\cuglrn\myscript"
    
```

### *Executing a Script from Visual Basic*

Use any mechanism (or any language, for that matter) to create a process that runs the above command line. The requirement is that the initial window state must be specified as hidden. This can be accomplished with the Visual Basic Shell function as follows.

⚠ *Warning! Although the following Shell function is a complete Visual Basic subroutine, it is not intended to recommend any particular programming style, nor is the code supported by Computron. Its purpose is to illustrate the format of the command line string and the required parameters to the Shell function. This is one example of running a script. Another example could use the CreateProcess Win32 API from any language that supports calling Windows APIs.*

```
Private Sub RunCtronScript()  
  
Dim taskid As Long  
Dim commandline As String  
  
Rem -- put the following assignment all on one line...  
commandline = "f:\ctrn\exec\_ctrn\_ntstart.exe  
-c ""Ctron MyConfigName""  
""c:\mksnt\sh.exe f:\ctrn\exec\cuglrun\myscript"""  
  
On Error Goto errors  
  
Rem -- vbHide ***MUST*** be specified...  
taskid = Shell(commandline, vbHide)  
  
Exit Sub  
  
errors:  
MsgBox( "Shell command failed!" )  
Resume Next  
  
End Sub
```

The process runs as the same user who launches the script (i.e., the user that ran the Visual Basic program that ran the script). This user, of course, must be a valid Computron user.





9. To implement your changes you must first stop and then restart the Windows Server, as follows:
  - a) Double-click the Service Management option in the left panel to display the service name in the right panel.
  - b) Right-click the service to display a shortcut menu.
  - c) Click the Stop command to shutdown the server. Notice that the status changes from Running to Stopped.
  - d) Right-click the service again, but this time click Start on the shortcut menu.
  - e) Make sure the Status changes from Stopped to Running.
10. Finally, rerun the script process. The script log will record all the values being passed by the scripts being run.

## Sample Scripts Log

```

e:/temp/scripts.log
+ [ -n E:/ctrn10/exec ]
+ . E:/ctrn10/exec/_ctrn_/functions/wusage
+ . E:/ctrn10/exec/_ctrn_/functions/csnatvol
+ alias stty=false
+ alias tput=false
+ alias autoload=false
+ alias csfixterm=true
+ [ -n amf ]
+ export LOGNAME=pse
+ [ -n ]
+
PATH=;E:/ctrn10/exec\_ctrn_;E:/ctrn10/exec\_ctrn_;;c:\winnt
;c:\winnt\system32;d:\mksnt;c:\ctrn10\oitnt;c:\ctrn10\exec\_c
trn_;C:\CTRONVB\exec\_ctrn_;;
+ export PATH
+ cspuparm ENTER GP0 BACKPROC=WRITECKB PAYRECS=001000
CURPRFNC=PROCESS BSFNC= FPROC=WRITICKF PFKEY=00
+ wrun writeckf

```

# **Chapter 5**

## **Computron Service Administration**

# 5.1 Computron Server Administrator

## *Introduction to Computron Server Administrator*

The Computron Server Administrator for Windows (ConfigMan.exe) is a powerful server configuration and management tool that enables you to configure, monitor and manage the Computron application servers running under the Windows operating system. The Computron Server Administrator can run locally on the Computron application server or it can run remotely on any Windows client (excluding '9x), as long as it has network access to the application server.

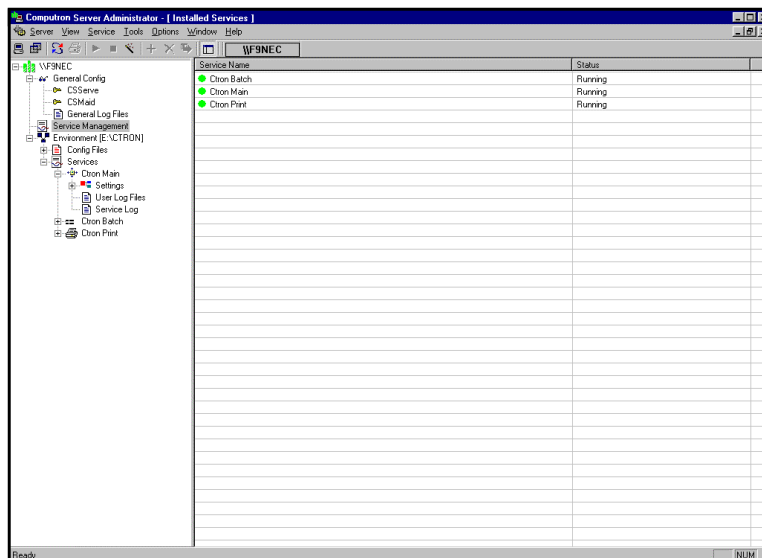
The Computron Server Administrator relies on Windows security to allow or disallow service maintenance activities, such as modifying configuration entries, printing log files or installing services. Additionally, the Computron Server Administrator provides an extra layer of security since it can be configured to restrict access to only those Computron users defined in a particular Windows user group.

## *Service Management*

The Computron Server Administrator is an executable file (CONFIGMAN.EXE) that resides in the \...\exec\\_ctron\_ directory (where \... represents the directory where the Computron software resides). Like all Windows executables, it can be launched via Explorer, by using the Start Run command or from the Computron Server Group on the Start menu (usually generated during setup). For easy access to the program, however, we suggest creating a shortcut icon on the desktop.

## Computron Server Administrator Window

The following sample shows how the Computron Server Administrator main window appears by default when launched.



**Main Computron Server Administrator open window**



The main window is divided into two panes:








- ◆ The service entries for the current machine are listed in an expandable tree format in the left pane.
- ◆ The details for a selected entry are listed in the right pane.






The panes can be resized by dragging the split bar to the left or right. The service items can be expanded to display any sub-items with either double-clicking or by clicking its plus sign (+) icon. Displaying an item's property list in the right pane requires only a single-click.

☞ *Note:* Print Config and Initiators (Batch) service items do not have property listings. A double-click on either of these items opens their respective dialog boxes.

### *Toolbar Buttons*

The following table describes each button on the main window toolbar.

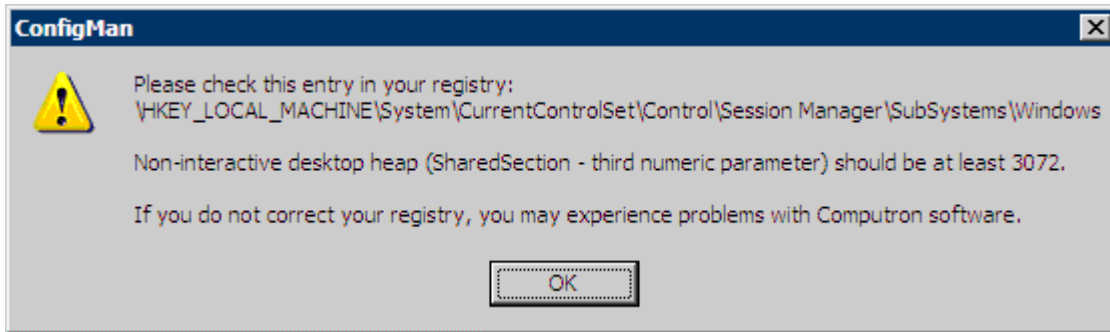
Button	Description
	Connect to a remote server.
	Start the Process Monitor utility.
	Refresh all list views; for example, Service Management list, Log File list, etc.
	Prints lists, log files, configuration files, etc.
	Start a service.
	Stop a service.
	Restart a service

Button	Description
	Install a service.
	Add a configuration item.
	Delete a configuration item.
	Modify a configuration item.
	Toggles between the default split windowpane or displaying only the installed service pane.

### Starting a Stopped Service

1. If the services are not displayed in the right panel, click Service Management in the left panel.
2. In the right panel, select the service that you want to start.
3. On the toolbar, click Start.
4. Confirm the Status change from Stopped to Running (from red to green).

☞ *Note:* When starting the main service, you get an information popup, illustrated below. In the Registry, the HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Control\Session Manager\Subsystems entry, the Windows parameter, set the third number (non-interactive heap) of the ObjectDirectory=\Windows SharedSection to 3072 or higher. Caution must be exercised with the number entered here, too high (about 8000) and Windows won't boot. This number, expressed in megabytes, determines the graphics usage of client machines.



**ConfigMan Information Message when Starting the Main Service**

## Stopping a Running Service

*Important!* Before stopping a service, use the Computron Process Monitor for Windows to verify that all users are logged off that service. For detailed instructions, refer to the “Computron Process Monitor” section in this chapter.

1. If the services are not displayed in the right panel, click the Service Management option in the left panel.
2. In the right panel, select the service that you want to stop.
3. On the toolbar, click Stop.
4. Click Yes to confirm your decision.
5. Confirm the Status change from Running to Stopped (from green to red).

## Restarting a Running Service

*Important!* Before restarting a service, use the Computron Process Monitor for Windows to verify that all users are logged off that service. For detailed instructions, refer to Section 5.11, “Computron Process Monitor.”

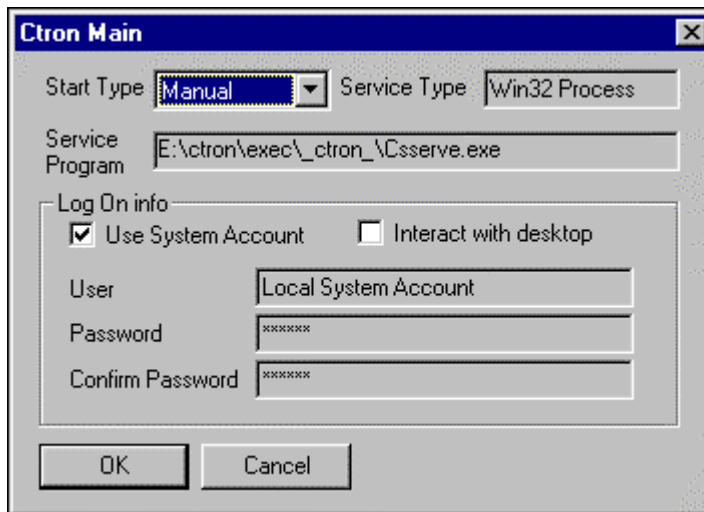
1. If the services are not displayed in the right panel, click the Service Management option in the left panel.
2. In the right panel, select the service that you want to restart.
3. On the toolbar, click Restart.

## Viewing and Setting Service Startup Properties

The Computron Server Administrator allows you to easily check the configuration properties of each installed service.

☞ *Important!* Some of the service properties are modifiable, but the service must be stopped and then restarted, for the changes to take effect. These options should only be modified under the direction of Computron support personnel.

1. If the installed services are not displayed in the right panel, click Service Management in the left panel.
2. In the right panel, select the service whose properties you want to check.
3. On the Service menu, Click Properties or right click on the service and select Properties. The Properties dialog box appears for that Service.



**Service Properties dialog box**

4. If you make any changes to the service, click OK to save them and return to the main window.

### *Service Properties Fields*

#### **Start Type**

This modifiable field allows you to select how the service is started. Valid values are:

Manual – Service must be manually started by a Computron Server Administrator.

Automatic – Service starts automatically when the Windows Server is rebooted.

Disabled – Prevents service from being started.

#### **Service Type**

This non-modifiable field is the type of service. (All Computron services have a service type of Win32 Process).

#### **Service Program**

This non-modifiable field is the full path of the service program.

## *Log On Info Panel*

### **Use System Account**

This specifies that the service logs onto the system account, rather than a user account. This is the recommended option.

### **Interact with desktop**

This specifies whether you want the service to provide a user interface on a desktop that can be used by whoever is logged in when the service is started. This option can be used only if the service is running as a Local System Account.

- ☞ *Important!* This option should always be turned off unless you are performing diagnostic activities under the direction of Computron support personnel.
- ☞ *Note:* This option must be turned on when enabling “COBOL Debug mode”. For more information about COBOL Debug mode, refer to Section 6.9, “COBOL Debugging.”

### **User**

This field enables you to assign a logon user account to a service. Although most services must log on to the system account, some services can be configured to log on to special user accounts.

- ☞ *Important!* All services should use the system account. Unless directed otherwise by Computron support personnel.

### **Password**

#### **Confirm Password**

These fields provide a place for you to type and confirm the password for the user account. This is the password that was assigned to the user account in the Windows User's and Groups.

- ☞ *Note:* The User and Password/Confirm Password fields generally display non-modifiable Local System Account information. Since this is the recommended option, these fields should only be modified under the direction of Computron support personnel.

## **Configuring Services**

Each Computron service has an associated set of named values called configuration items. Among other settings, these configuration items specify values like directory names, TCP/IP port numbers and logging levels. Computron services may be configured by modifying their corresponding set of configuration items.

The Computron Server Administrator provides an easy way to maintain these settings, thereby eliminating the need to run the Windows Registry Editor. Keep in mind, however, that many configuration items have default values that should not be modified unless directed to do so by a Computron representative. These items were assigned appropriate values when you ran Setup to initially install the software.



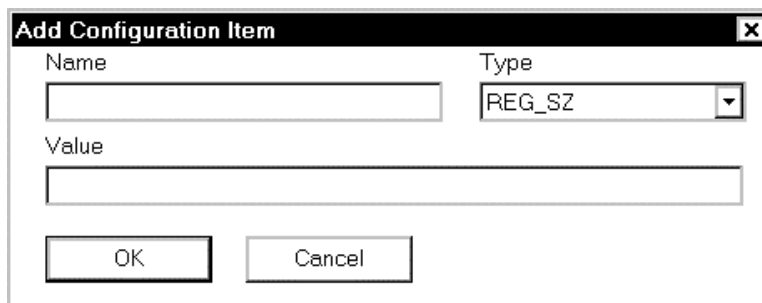
- Communication – port numbers and Validate/Impersonate settings.
- Switches – internal switch settings.
- Debug – debug settings.
- Misc – other configuration settings.

## Adding Configuration Items



☞ *Important!* Whenever configuration items are added, the Service must reread all settings. This may require stopping and restarting the modified service.

1. On the toolbar, click Add.



**Add Configuration Item dialog box**

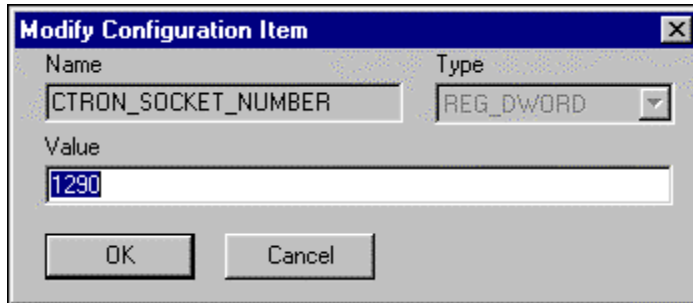
2. Enter the following information for the item:
  - ◆ Name – the name of the configuration item.
  - ◆ Type – the type of data:
    - REG\_SZ – a simple string.
    - REG\_EXPAND\_SZ – a string that contains unexpanded references to environment variables (i.e. “%CTRON\_ROOT”).
    - REG\_DWORD – a 32-bit number.
  - ◆ Value – the value of the configuration item.
3. Click OK.
4. The configuration item has been added to the service.

## Modifying Configuration Items



☞ *Important!* Whenever configuration items are modified the Service must be stopped and restarted for the changes to take effect.

1. Select the configuration item (right panel) to be modified.
2. From the toolbar, click Modify.



**Modify Configuration Item dialog box**

3. The following information is read-only for the configuration item:
  - ◆ Name – the name of the configuration item.
  - ◆ Type – the type of data.
4. Modify the Value box for the item.
5. Click OK.
6. The item has been modified.

### *Deleting Configuration Items*



**Important!** Whenever configuration items are deleted the Service must be stopped and restarted for the changes to take effect.

1. Select the configuration item (right panel) to be deleted.
2. From the toolbar, click Delete.
3. Click OK to confirm deletion process.
4. The item has been removed.

## Applying changes made to the configuration items

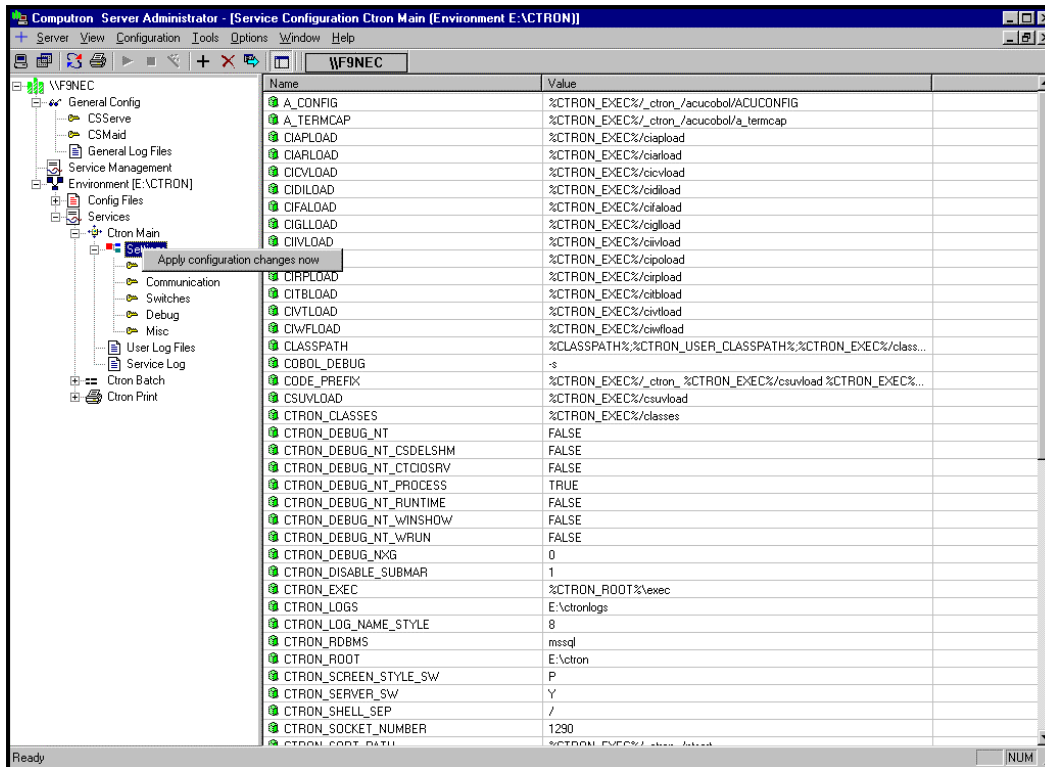
After adding, modifying or deleting a configuration item, it is necessary for the service to reread all of its settings.

**Warning!** *The following steps only apply to a modified CSServe Service! All other services (e.g. Print, Batch, etc.) must be restarted.*

1. From the service item tree in the left panel, expand (by double-clicking) Services.
2. Double-click the desired Service Name to display its sub-items (e.g., CSServe is named Ctron Main in the illustration that follows).



- Right-Click on Settings to display “Apply configuration changes now”, as illustrated here:



Apply configuration changes now dialog box

- Left mouse click on “Apply configuration changes now.”

*Note:* This forces the service to reread all configuration settings. The changes apply to all clients that login after the reread.

**Warning!** *Not all changes can be applied using this method. If the changes do not affect newly connected clients, you must log all clients off and restart the modified CSServe service.*

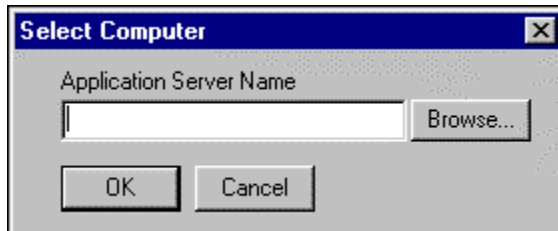
*Important!* If it necessary to apply the modifications to clients already connected, all clients are required to disconnect from the CSServe service before restarting the service.

## Connecting to a Remote Machine



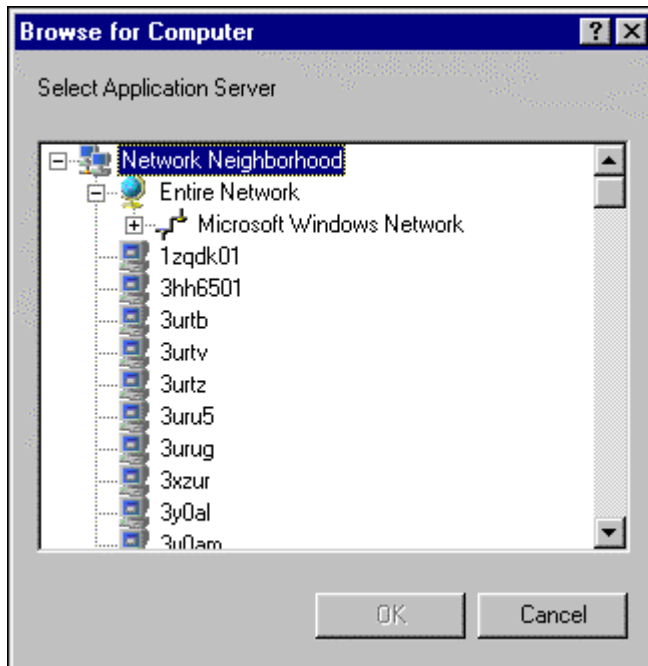
If the Service that you need to maintain is installed on another networked Windows machine that you have access to, you can log onto it via the main Computron Server Administrator window.

1. From the toolbar, click Connect to Server.



Select Computer dialog box

2. Enter the remote machine's name in the Application Server Name field or click Browse to locate it via the following dialog box:

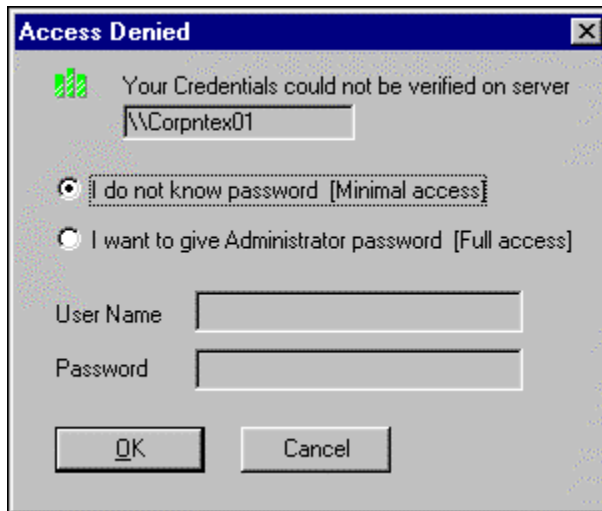


Browse for Computer dialog box

3. Select computer from the given list by clicking on it. Click OK.
4. Click OK. You have now returned to the Computron Server Administrator open window.
5. Without proper access credentials, the following window appears:

*Note:* If you have the proper access credentials, the computer name you selected appears at the end of the toolbar, with its installed services listed in the Service Management list. If you do not have the proper access credentials, you can still view

the list of services (and their status) but you cannot start or stop them. You also cannot modify any configuration information.



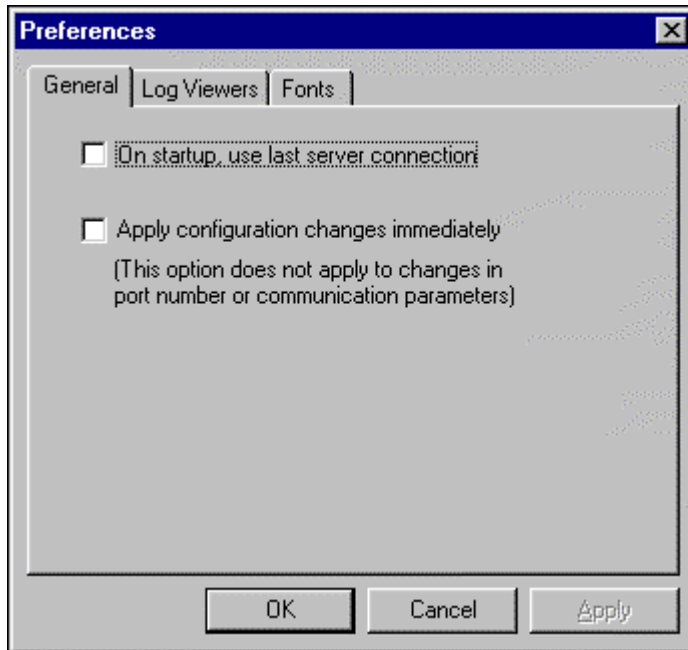
**Access Denied dialog box**

6. To request full access to the machine, select the radio button, I want to give Administrator password (Full Access), and then enter a valid User Name and Password. Click OK.
7. Upon entry of a valid User Name and Password, you are granted access to the Service Management tools.

## Automatic Remote Logon Option

This option allows you to return to the server last used in your previous session.

1. From the View menu, click Preferences, and then click on the General tab.
2. Select the On Startup, use last server connection check box.



General tab dialog box

3. Click OK.

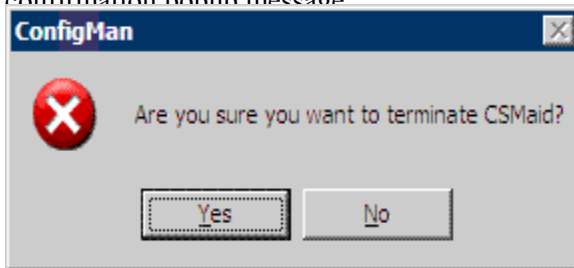
## Shutting Down the CSMAid Process

The CSMAid process manages the creation, access and removal of various Computron objects, as required. The Computron Server Administrator provides a Terminate CSMAid command, which can be used to shut down the CSMAid process. This action must be performed when the system is rebooted for disaster recovery purposes, when installing a Computron patch or before installing a new version of CSMAid.

**Warning!** *Computron does not recommend stopping the CSMAid process while users are connected to the system or while other Computron services are running. Therefore, first use the Computron Process Monitor to verify that all users are logged off the system before proceeding with the following instructions. For more information about the Process Monitor, refer to Section 5.11, "Computron Process Monitor."*

1. If necessary, open the Computron Server Administrator.
2. Click Service Management in the tree to activate the list of Services in the right panel.

3. Stop all Computron services. CSMaid cannot be stopped before all the other services are stopped.
4. From the Service menu or right-click on the CSMaid service, click the Stop command. You get a confirmation popup message.



**ConfigMan Confirmation Message for CSMaid Service Stop**

5. Click Yes when prompted to confirm termination of CSMaid.
  - ☞ *Note:* After shutting down CSMaid, the Computron Process Monitor cannot access active process information until CSMaid is restarted. In the case of an emergency shutdown and restart, CSMaid is automatically restarted upon restarting CSServe, which is usually restarted upon reboot. CSServe only restarts automatically if the server's Start Type property is set to Automatic. For more information, see "Viewing and Setting Service Startup Properties" earlier in this section.

## Editing Configuration Files

The Computron Server Administrator enables easy access to the general (server) configuration and service configuration files for all environments, so that you don't have to spend time searching through subdirectories to find them.

⚠ *Warning!* These files should only be modified under the direction of Computron support personnel.

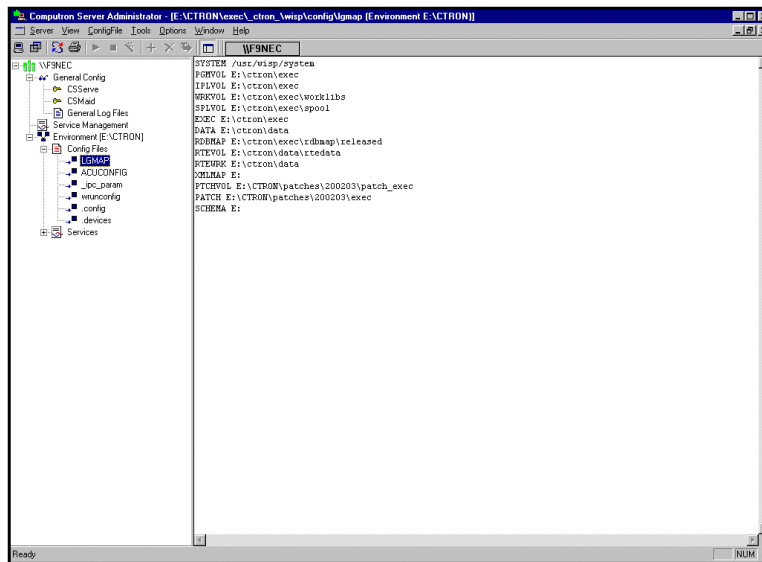
## Environment Files

The following configuration files are available for each Computron environment installed on the server.

- ◆ LGMAP file – maps volume names used by Computron systems to a proper physical location within the Windows file structure (i.e., paths). Volume names are then used by Database Map File Maintenance and other functions. For more information about the LGMAP, refer to Appendix C, "LGMAP File."
- ◆ AUCONFIG file – controls internal ACUCOBOL configuration.
- ◆ \_ipc\_param file – contains default entries for various Hold List configuration items.
- ◆ wrunconfig file – contains various parameters necessary for the execution of application programs.

- ◆ .config – contains overrides for certain default parameters, mostly database and some localization.
- ◆ .devices – contains local device definitions. This file is used internally by the runtime.
- ◆ .last\_patch – contains the date (YYYYMM) of the original full install and the date of the latest patch.

To access the Environment files, in the left panel, double click the Config Files entry listed directly under the Environment entry that you are working in. Upon clicking one of the Config Files entries (e.g., LGMAP), its contents are displayed in the right panel. The following sample shows the contents of the LGMAP file in the right panel:



Sample LGMAP File

When a configuration file is displayed, you can perform the following tasks:

- ◆ Edit the file by typing in the right panel.
- ◆ Cut, Copy or Paste text via the ConfigFile, Edit menu.
- ◆ Print or Save the file via the ConfigFile menu.

☞ *Note:* When making changes, be sure to Save the file for the changes to be put into effect.

## General Configuration Items

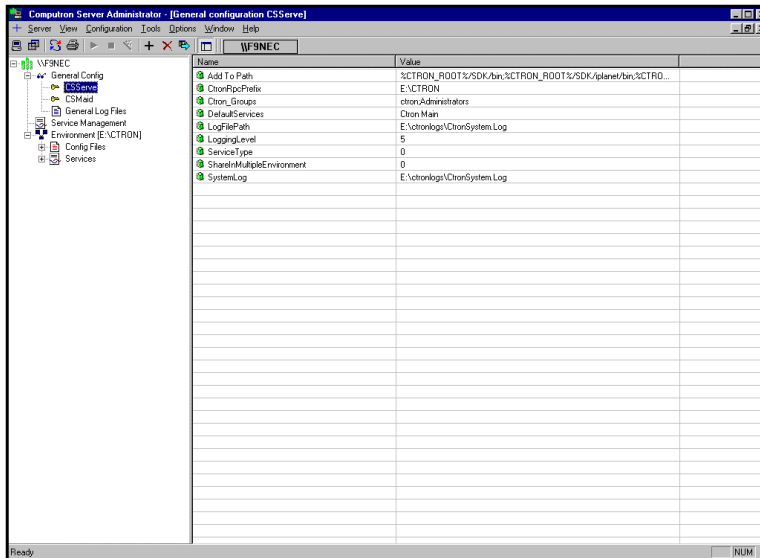
The general configuration items are universal for all Computron environments installed on the server.

☞ *Important!* These items should only be modified under the direction of Computron support personnel.

- ◆ CSServe – this connection service listens for connection requests from Computron clients and creates application processes as necessary.

- ◆ CSMaid – manages the creation, access and removal of various Computron objects, as required.

To access the general configuration values, in the left panel double-click the General Config entry (listed directly under the Server Name). Upon clicking one of the General Config entries (e.g., CSServe), its configuration values are displayed in the right panel. The following sample shows the CSServe configuration values in the right panel:



Sample CSServe File

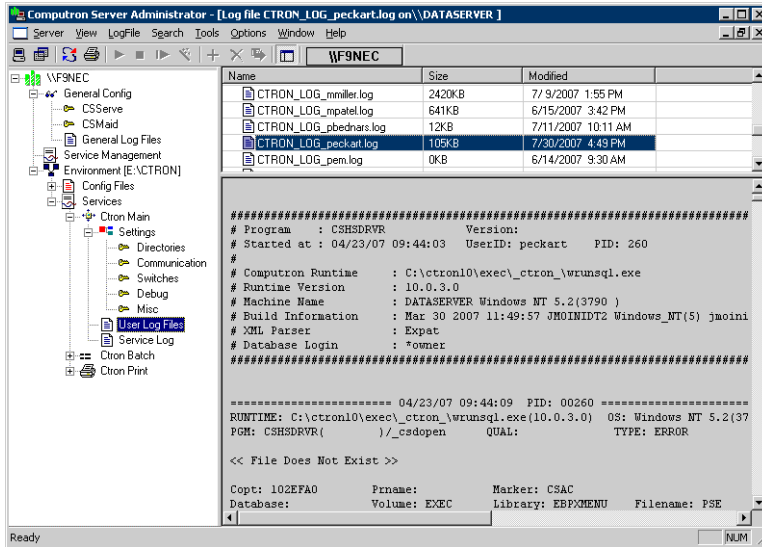
When general configuration values are displayed, you can perform the following tasks:

- ◆ Print the values via the Configuration menu.
- ◆ Add, Modify, or Delete the configuration values via the Configuration, Config Item menu.

## Managing User Log Files

The Computron Server Administrator enables easy access to all service log files and user log files for all environments, so that you don't have to spend time searching through subdirectories to find them.

1. From the service item tree in the left panel, expand the Services entry for the Computron Environment that you want to access.
2. Double-click the desired Service Name to display its sub-items (e.g. CSServe is named Ctron Main).
3. Click the User Log Files option to display all the log files for that service in the right panel, sorted by user ID.
4. Double-click any log file entry (in the upper right panel) displaying the contents of the file in the lower half of the right panel, as illustrated here:



**Sample Log Files**

- Once a log file is displayed you can perform any of the following tasks:
  - View, Truncate, Delete, Send via E-mail or Print the file via the LogFile menu.

## Truncating Your Log Files

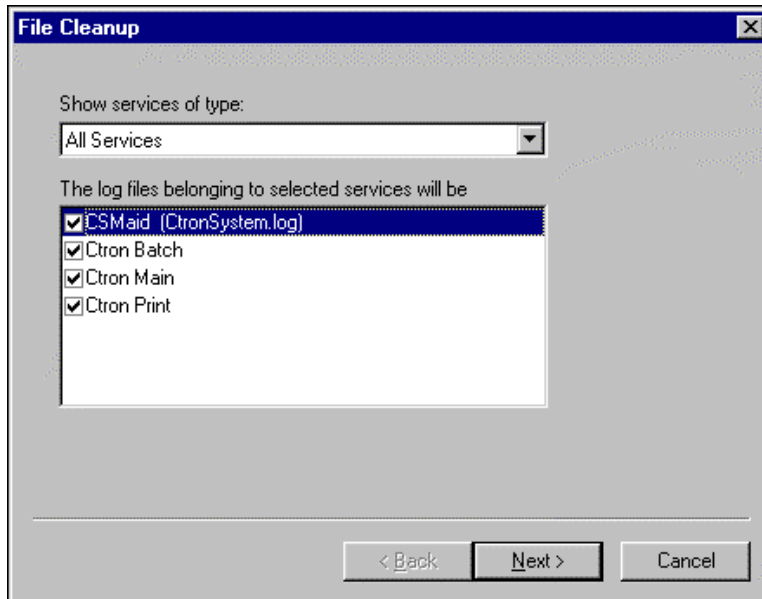
Since files and log entries are continually accumulating on the Computron server, it may become necessary to clean up those files and logs. If left unattended, the logs can become bloated and unneeded work files can be left in place, all consuming disk space. To help alleviate this, the Computron Server Administrator includes a File Cleanup option. This option allows you to automate the cleanup cycle frequency (e.g., every 5 days), select the time of day at which to perform this operation, choose the length that you want your log files truncated to, and select which log files to truncate and which unneeded work files to delete.

Manual Log File Truncation allows the System Administrator to truncate any log file immediately.



## *Automated Log File Truncation*

1. On the Option menu, click File Cleanup.



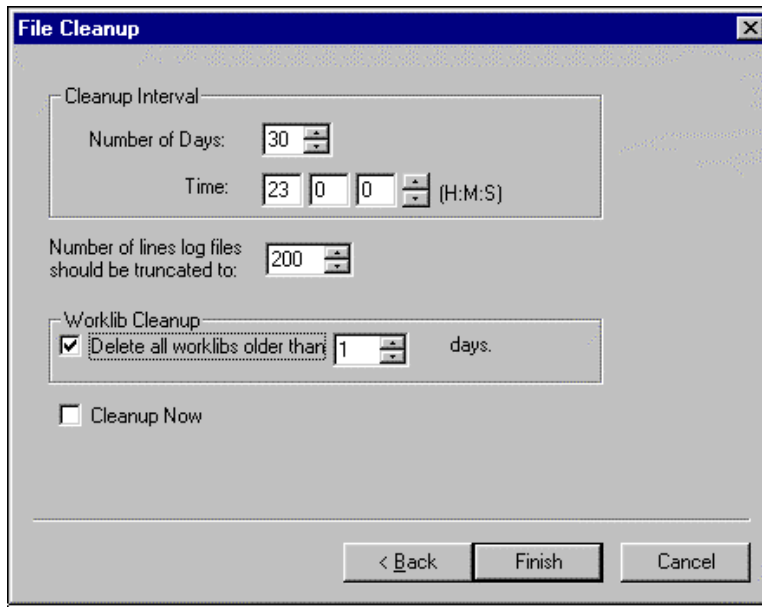
**Select services to clean**

2. Select the services that need cleanup, by placing a checkmark in the appropriate box.

*Note:* It is possible to filter the services to be managed, by selecting the appropriate service type from the pull down menu.

- ◆ **CSServe Services** – Displays only the CSServe type services installed on the server. This allows management of user log files (i.e. CTRON\_LOG\_username.log).
- ◆ **CSMaid Service** – Allows the management of CSMaid log called CtronSystem.log. (There is only one CSMaid per server).
- ◆ **Print Services** – Displays only the Print type services on the server. This allows management of Print log files called CtronPrint.log.
- ◆ **Batch Services** – Display only the Batch Queue type services installed on the server. This allows management of Batch logs called CtronBatch.log.
- ◆ **Hold Services** – Display only the Hold type services installed on the server. This allows management of Hold logs called CtonHold.log.

3. Click Next.
4. Select your “Cleanup Interval”.



Select interval of cleanup

### Number of days

Enter the frequency (in number of days) that you want the truncate operation to take place.

### Time

Enter the time (hour, minute, and second) at which you want to the truncate operation to take place.

*Note:* Use 24-hour time notation.

### Number of lines log files should be truncated to

Enter the amount of lines that you want to truncate the selected log files to.

*Note:* The User Log Files are truncated to the nearest line break entry (=====), which indicates the beginning of a log entry. If the number of lines specified causes the truncation to fall in the middle of a log entry, the truncation point is moved to the beginning of that entry.

### Delete all worklibs older than ... days

This selection deletes worklib directories that are older than a selected interval. This helps CSMaid clean up unused directories that may have been missed during CSMaid's normal cleanup cycle.

### Cleanup Now

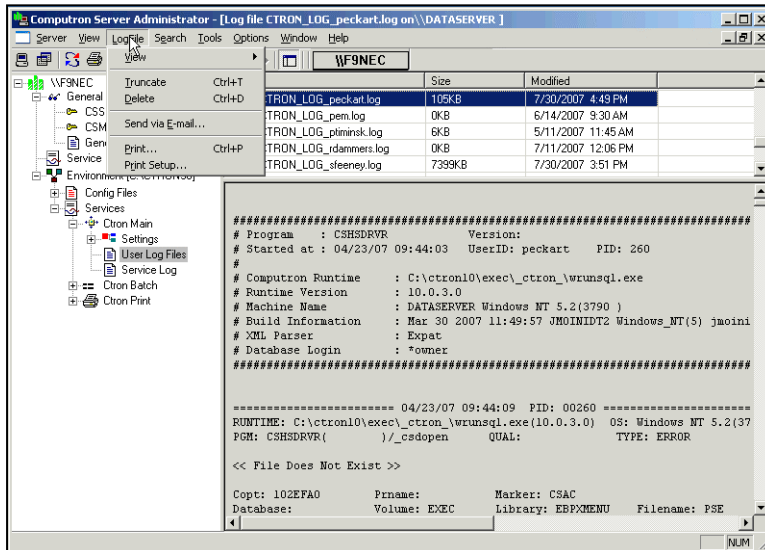
Select this box to perform the cleanup at the beginning of the next CSMaid cycle. If this check box is clear, the cleanup operation occurs during the next cleanup cycle of CSMaid after the selected time.

5. Click Finish to exit and save your log file cleanup settings.

Note: Since the default cleanup interval for CSMaid is every ten minutes, file cleanup usually occurs within ten minutes of the selected time (during CSMaid's next cleanup cycle). If the cleanup interval is altered, the file cleanup time may be altered as well.

## Manual Log File Truncation

This process removes the user log file contents. The log file remains in the directory, but its size is 0.



### LogFile management menu

1. On the LogFile menu, click Truncate.
2. Click Yes to confirm the truncation process.
3. The log file has been truncated.

**Warning:** Using this function removes all log file content. Once deleted, this content is unrecoverable. For a partial truncation, use automated Log File Truncation.

## Printing Your Log Files

1. On the LogFile menu, click Print.
2. Check settings in the Print Dialog box. Click OK.
  1. Your log file has been printed.

## Emailing Your Log Files

1. On the LogFile menu, click Send via E-mail.
2. Computron opens an email message window and includes your file as an attachment.

3. A message dialog box appears with the log file attached.
4. Fill in the names of the recipients in the To, Cc, and Bcc boxes of the message.
5. Send the message.

## Deleting Your Log Files

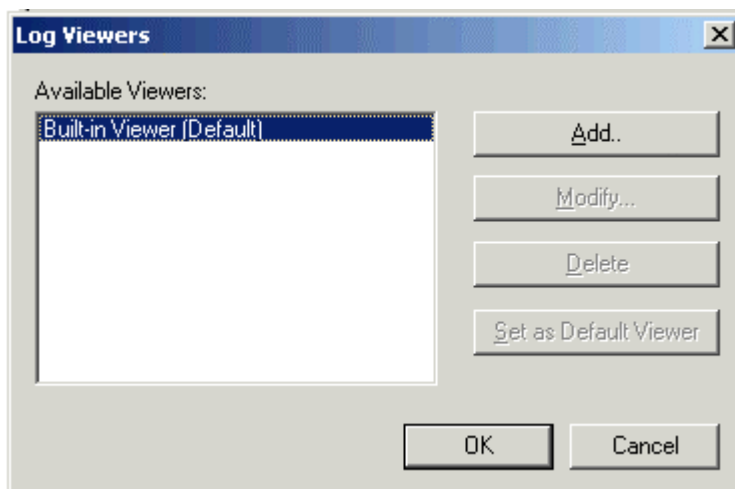
1. On the LogFile menu, click Delete.
2. Click Yes to confirm deletion process.
3. The user log file has been removed from the directory.

## Editing Log Viewer

Although the Computron Server Administrator includes a default Built-in Viewer for accessing your log files, it also provides the option to select an alternate viewer. For example, you may be comfortable using Notepad as your default log viewer.

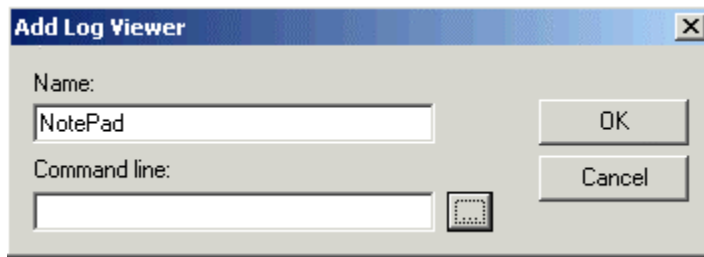
### *Add a Log Viewer*

1. From the Options menu, select Log Viewers.



**Log Viewers dialog box**

2. Click the Add button.



**Add Log Viewer dialog box**

3. Enter the following information for the log:
  - ◆ Name – the name of the viewer (e.g., Notepad).
  - ◆ Command Line - the command line for the viewer (e.g., notepad.exe). You can browse for the command.
4. Click OK. The viewer has been added to the viewer list.

### *Delete a Log Viewer*

1. From the Options menu, select Log Viewers.
2. Select the log viewer to be deleted.
3. Click the Delete button.
4. Click OK to confirm deletion process.
5. The log viewer has been removed.

### *Modify the Log Viewer*

6. From the Options menu, select Log Viewers.
7. Click the log viewer to be modified.
8. Using the Modify button, change the Command line of the Log Viewer.
9. Click OK.

### *Set the Default Log Viewer*

1. From the Options menu, select Log Viewers.
2. Click the log viewer to be set as the default.
3. Click the Set as Default Viewer button. Another line is added that says Built-in Viewer (Default). The viewer that was the original log viewer selected also has (Default) set next to it.

## *Override the default Log File Viewer*

This setting is temporary and only applies to the selected viewer. It does not alter your default log viewer. To alter the default log file viewer, see “Set the Default Log Viewer” earlier in this section.

### **User Log File**

1. From the tree in the left panel, expand the Services entry for the Computron Environment that you want to access.
2. Double-click the desired service name to display its sub-items (e.g., Ctron Main).
3. Click User Log Files.
4. In the right panel, select the user log file to be viewed, right click.
5. Click on View. Select the viewer in which to display the log.
6. The User Log File now appears in the chosen viewer.

### **Service Log**

1. From the tree in the left panel, expand the Services entry for the Computron Environment that you want to access.
2. Double-click the desired service name to display its sub-items (e.g., Ctron Main).
3. Highlight the Service Log, right-click.
4. Click on View. Select the viewer in which to display the log.
5. The Service Log now appears in the chosen viewer.

## 5.2 Computron Batch Queuing System

### *Introduction to the Batch Queuing System*

The Computron Batch Queuing System is a service that controls and monitors the submission and execution of background jobs within the Computron system. It works in conjunction with the CSServe service to control the number of jobs that can run concurrently in background mode and the priority at which those jobs run.

The following instructions refer to two servers, the “Application Server” and the “Remote Server.” The Application Server (the primary server) is the machine that users connect to, to log on interactively, run programs and launch background jobs. The Remote Server is the machine where background jobs are actually executed. Users do not connect to this machine to log on interactively.

### *Additional Considerations for Batch Queuing System*

To utilize a distributed batch queue configuration, a relational database (MSSQL or Oracle) must be used. Computron does not support Vision for use with distributed batch queuing.

### What is Background Processing?

Background processing allows you to submit processes to a job queue so that they can be run at a later time – without tying up user workstations. A batch queuing system typically limits the number of background jobs that may run simultaneously. If the configured maximum number of background jobs is already running, any additional jobs must wait in the job queue until they are called. This ensures that system resources are not over-committed and that system responsiveness and throughput are maintained. To further optimize system throughput and user productivity, you can configure background jobs to run at a lower priority, thereby giving your interactive jobs a performance advantage over background jobs.

The Computron Batch Queuing service combines the basic functionality described above with the following customization features:

- ◆ Configurable overall maximum number of jobs that can run simultaneously;
- ◆ Configurable maximum number of jobs that can run simultaneously per job class;
- ◆ Configurable process execution priorities per job class (run priority);
- ◆ Ability to change the ordering of jobs in the job queue via the job queue priority;
- ◆ Jobs are run in the security context of the user that originally submitted the job;
- ◆ All jobs entering and leaving the job queue are fully logged;

### Introduction to How the Batch Queuing System Works

This section provides a high-level overview of the major concepts related to Computron’s Batch Queuing System. The intent is to provide a reasonably complete description of the major components and functional aspects of the system without overwhelming you with details. To help put things in perspective, the sections following this one describe the Batch Queuing System from a configuration and usage point of view.

The Computron Batch Queuing System runs as a native Windows service. It is installed and configured using the Computron Server Administrator utility. Computron application programs submit jobs by communicating with the Computron Batch Queuing service. The service is responsible for placing jobs on the queue, maintaining security information, and executing jobs. It also services requests that come from the Computron Process Monitor, a utility that allows you to view and control both the queued and running jobs. The Process Monitor sends requests to the service to enumerate, change, hold, release and delete jobs on the job queue. The Process Monitor provides a GUI front-end to these Batch Queuing System service functions.

☞ *Technical Note:* The service registers as a Remote Procedure Call (RPC) listener. Other Computron processes talk to the service by making RPC calls.

☞ *Note:* To minimize CPU utilization, all threads wait on events (i.e., go to sleep) when there is no work available to be performed.

### *The Job Queue*

The Batch Queuing System provides a single job queue that receives all jobs that are submitted to background. The entire job queue can be in a Held or Released state. When the job queue is held, no jobs on the queue are run. When the job queue is in its released (normal) state, jobs on the queue are run.

☞ *Note:* Jobs are run in accordance with the Computron Batch Queuing System's job scheduling rules, which are explained later in the "Run Priority" and the "Job Queue Priority" sections.

The service uses an in-memory table to keep track of jobs on the job queue. Additionally, a job file exists on disk for each job. Most of the job's attributes, such as the name of the program to run and environment information, are stored in the job file. The other attributes, such as job queue priority and security information, are stored in the in-memory table. Because the security information cannot be stored on disk, the Batch Queuing System flushes all remaining jobs from the job queue when the service is stopped for any reason.

### *Initiators*

A job resides on the job queue until an initiator runs it. Initiators run jobs by finding an available job on the queue, removing that job from the queue and then running it. Each initiator can run one job at a time. While a job is running, the initiator that started the job waits. When a job finishes, the initiator that started the job looks on the job queue for the next job that is eligible to be run by that initiator. If it finds a job to run, it runs the job. It then waits for the job to complete and then looks for another job to run, until the service is stopped.

If an initiator is free (i.e., not currently running a job) and there are no jobs for it to run, it goes into sleep mode until a job is available for it to run. Since an initiator must run each background job, the number of configured initiators determines the maximum number of background jobs that can run simultaneously.

☞ *Technical Note:* A thread in the service implements each initiator.

### *Job Class*

Each job on the job queue has a job class attribute. Correspondingly, each initiator can be configured to only accept jobs of a particular job class (or to accept jobs of any class). A job class is a one-character



designator that allows different types of jobs to be channeled to different sets of initiators. By assigning a certain number of initiators to a particular job class, you control the maximum number of jobs of that each job class can run simultaneously. By configuring the run priorities of those same initiators (as described in the next section), you control the execution priority of all jobs of a particular job class.

### *Run Priority*

Each initiator is assigned a run priority. All jobs run by a particular initiator are executed at that initiator's assigned priority. Typically, you should configure all initiators for a particular job class to have the same run priority. The run priorities are Lowest, Below Normal, Normal, and Above Normal (from lowest to highest). However, we recommend using only the Lowest and Below Normal settings.

### *Job Queue Priority*

Each job on the job queue has a queue priority. This priority controls the sequencing of jobs on the job queue. Jobs with a higher queue priority are removed from the job queue and run before jobs with a lower job queue priority. Jobs of equal queue priority are processed in arrival sequence. To help manage the sequencing of jobs on the job queue, system administrators can use the Computron Process Monitor to change queue priorities. All jobs are submitted with a queue priority of 5, which is the middle range of allowable queue priorities. This allows administrators, on an exception basis, to move individual jobs higher or lower than the rest of the jobs in the queue. Higher numbers indicate higher priorities. For example: a job with a queue priority of 6 runs before a job with a queue priority of 5.

### *Job Queue Status*

Each job on the job queue has a job queue status attribute. The job's status may either be Ready or Held. A status of Ready means that the job is run by an initiator when it's that job's turn to run (based on job class, run priority, arrival sequence and initiator availability). A status of Held means that the job remains on the job queue indefinitely, even if an initiator is available to run the job. A Held job must be released before it can be run (releasing a Held job changes its status to Ready).

☞ *Note:* Before the installation, you might benefit from the information in the “Configuring the Batch Queuing System” at the end of this section

## ***Installing the Batch Queuing Service***



The Computron Server Administrator features a Services Install Wizard that enables you to establish a Computron Batch Queuing Service. The following instructions explain how to install and configure your service.

The Computron Server Administrator also allows you to install additional Batch Queuing services. To make installation of additional services easy, it also provides an option to use an existing configured Batch Queue Service as a template. This is useful when creating diagnostic versions of services where only a few parameters need to be changed.

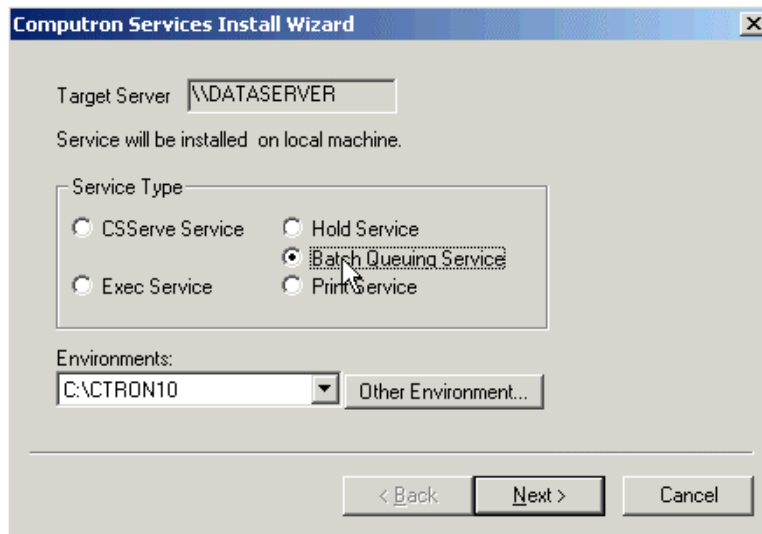
☞ *Note:* Fully read the Batch Queuing Reference Section at the beginning of this section, to gain a basic understanding of the Batch Queuing System. It helps you determine how to best set up the Batch Queuing System to suit your needs.

## Installation of Batch Queue – Application Server Only

☞ *Note:* For instructions on systems using distributed batch, see “Installation of Batch Queue Service – Application Server with Distributed Batch Server” later in this section.

1. Open the Computron Server Administrator window.
2. From the toolbar, click Install Service to display the first Install Wizard dialog box.

☞ *Note:* The Computron Service Management list must be active in order to access the Install Wizard.



Computron Services Install Wizard dialog box

Enter the following information in the dialog box:

### Service Type

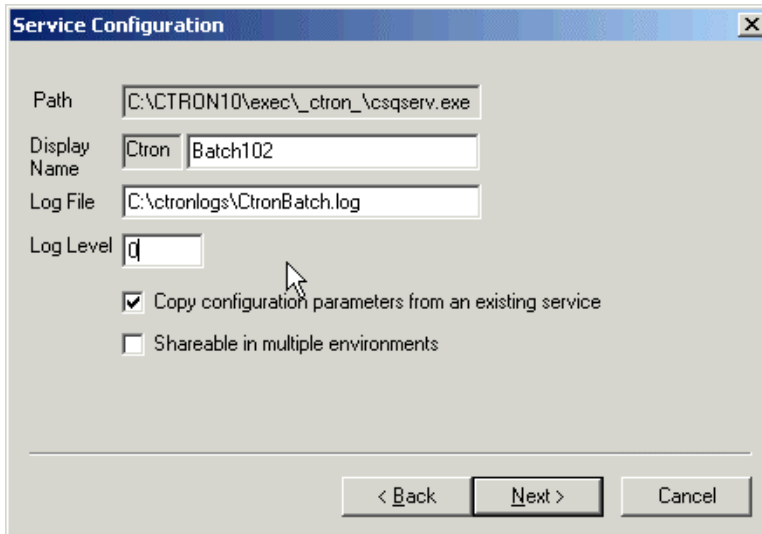
Choose Batch Queuing Service.

### Environments

Select the environment where you are installing the service.

☞ *Note:* If the environment is not listed, click the Other Environment button to browse for the cssrvctl.cfg configuration file for that environment. This would happen only if there were no configured services in a particular environment.

3. Click Next.
4. The Service Configuration box appears.



The image shows a 'Service Configuration' dialog box with the following fields and options:

- Path:** C:\CTRON10\exec\\_ctron\_\csqserv.exe
- Display Name:** Ctron Batch102
- Log File:** C:\ctronlogs\CtronBatch.log
- Log Level:** 0
- Copy configuration parameters from an existing service
- Shareable in multiple environments

Buttons at the bottom: < Back, Next >, Cancel.

**Service Configuration dialog box**

Enter service information in the following fields:

**Display Name**

Enter a name for the new service.

**Log File**

This displays the default file name and location for the service log.

**Log Level**

This defaults to 0. Valid values are 0–5, where 0 is normal logging; 1–5 are increased levels of diagnostic logging.

**Copy configuration parameters from an existing service**

Select this check box if you want to use the configuration parameters from an existing Batch Queuing Service type to create a new service. This option is useful when creating diagnostic versions of services where only a few parameters need to be changed.

**Shareable in multiple environments**

This check box should normally be cleared unless otherwise directed by Computron support personnel.

The remaining informational field on this window is non-modifiable.

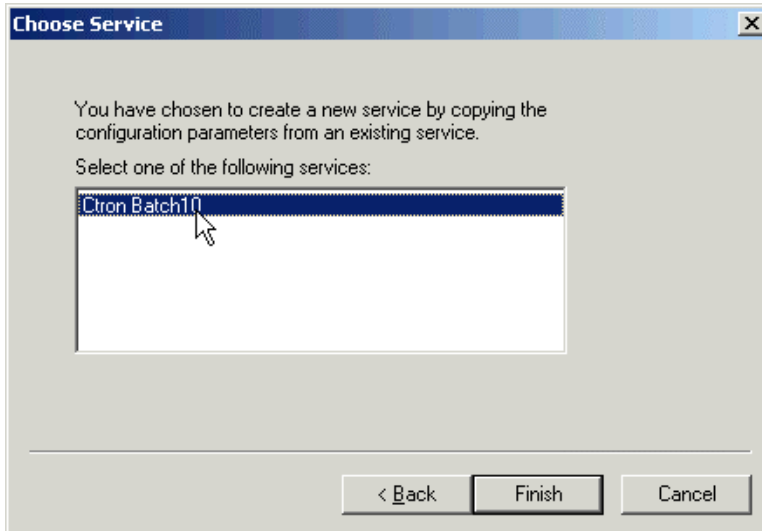
**Path**

This is the path to the executable file for the Batch Queuing service. If the path doesn't match the entry you selected in the Environments box on the previous window, double-check the cssrvctl.cfg file to make sure the CTRON\_ROOT entry is correct.

5. Click Next.

### *Using Copy Configuration Parameters*

- ☞ *Note:* If not using the Copy configuration parameters from an existing service feature, proceed to step 8.
- If you selected the Copy configuration parameters from an existing service check box, the Choose Service dialog box is displayed. Select the Batch Queuing Service whose configuration parameters you want to copy.



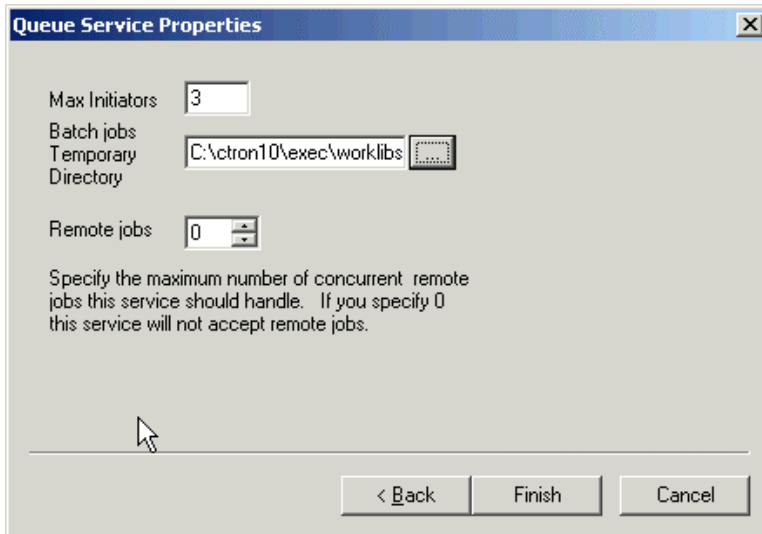
**Choose Service dialog box**

- Click Finish.

Computron Batch Queuing Service – Application Server Only installation is complete. The new batch queuing service is now listed in the Service Management list.

### *Not Using Copy Configuration Parameters*

- If not using the Copy Configuration Parameters from an existing service feature the Queue Service Properties dialog box is displayed.



**Queue Service Properties dialog box**

Enter the following information:

**Max Initiators**

Enter the number of Batch Queuing initiators that are configured for the service during installation or accept the default value of three. This number defines how many batch jobs can run at the same time.

**Batch jobs Temporary Directory**

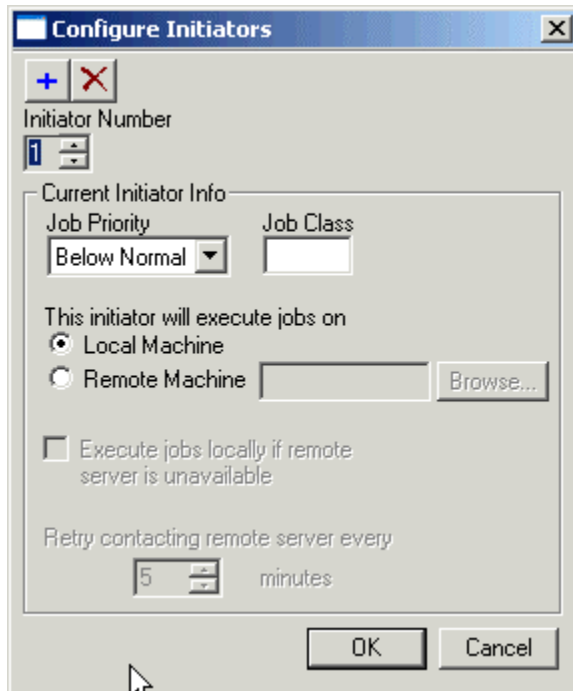
Use the browse button to locate the temporary work directory (Joblib). A typical directory location is C:\ctron\exec\worklibs\Joblib.

**Remote jobs**

Set this to zero unless using a distributed batch server. For instructions on systems using remote servers, see “Installing a Batch Queuing Service with a Distributed Batch Server.”

9. Click Finish.
10. If prompted to create a directory, verify the destination and click Yes to proceed.

## 11. Configure Initiators.



**Configure Initiators Dialog Box**

12. Complete the following information for each initiator. Failure to customize an initiator results in default values.

### **Initiator Number**

Use the Up/Down arrows next to this field to identify the initiator to be configured. This field is used to select the initiator number to be configured. The default value is derived from the number of Max Initiators entered on the Queue Service Priorities window.

### **Job Priority**

Choose a job priority of Above Normal, Normal, Below Normal, or Lowest (from highest to lowest). Default value is "Below Normal."

☞ *Note:* Computron recommends using the Below Normal priority. See "Configuring the Batch Queuing System/ Setting Run Priorities" in this section for more details.

### **Job Class**

Enter a one-character designation of the job class (e.g., A, B, C, etc.) or leave it blank for a blank job class. This initiator only processes batch jobs of this job class. Default value is blank.

☞ *Note:* For information on using job classes effectively, see "Configuring the Batch Queuing System/ Job Classes" in this section for more details.

### **Local Machine**

Select this option for application server use only. This is a default value.

**Remote Machine**

Select this option if a distributed batch server is being used. For instructions on systems using remote servers, see “Installation of Batch Queue Service – Application Server with Distributed Batch Server.”

**Execute jobs locally if remote server is unavailable**

This option is only available if using a remote server. For instructions on systems using remote servers, see “Installation of Batch Queue Service – Application Server with Distributed Batch Server.”

**Retry contacting remote server every \_\_\_\_\_ minutes**

This option is only available if using a remote server. For instructions on systems using remote servers, see “Installation of Batch Queue Service – Application Server with Distributed Batch Server.”

13. Click OK. The initiators have been configured.

Computron Batch Queuing Service installation is complete. The new Batch Queuing service is now listed in the Service Management list.

## Installation of Batch Queue Service – Application Server with Distributed Batch Server

Distributed Background Processing is an option in Computron Windows Server Software which allows background jobs to be submitted to a separate server from where interactive processes are run. This frees up the processing power on the main application server so that it can focus on interactive jobs, thus giving better performance to interactive users. Background jobs run on a completely different server. This second server must be set up with the same version and patch level of Computron Software as the main application server, as well as the same database maps.

To use the Distributed Batch Server option (dbatch), you must be using a relational database (MSSQL or Oracle). Vision is not supported.

### *Preparatory work*

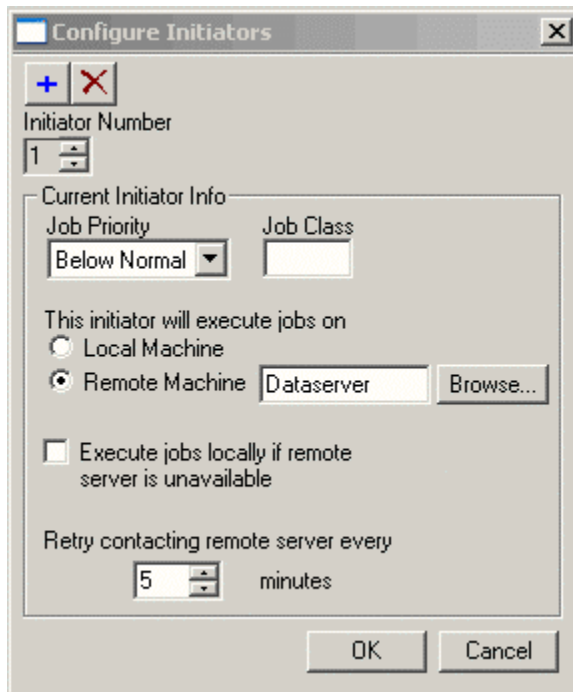
Install Computron Software on both the application server and the remote server.

- ◆ Ensure that the password and default database maps for the ctron sys account on both servers are the same.
- ◆ Start CSServe services (i.e. ctron main) on both servers.
- ◆ Test both installations by logging on as ctron sys and executing a program. Preferably, run a report in the Foreground mode.

### *Installation – Application server*

Run the installation as described earlier in this section. Follow steps 1 to 11.

12. Complete the information for each initiator. If you don't customize an initiator default values are used. For the definition of the fields on this window, refer to step 12 under the Not Using Copy Configuration Parameters heading earlier in this section.



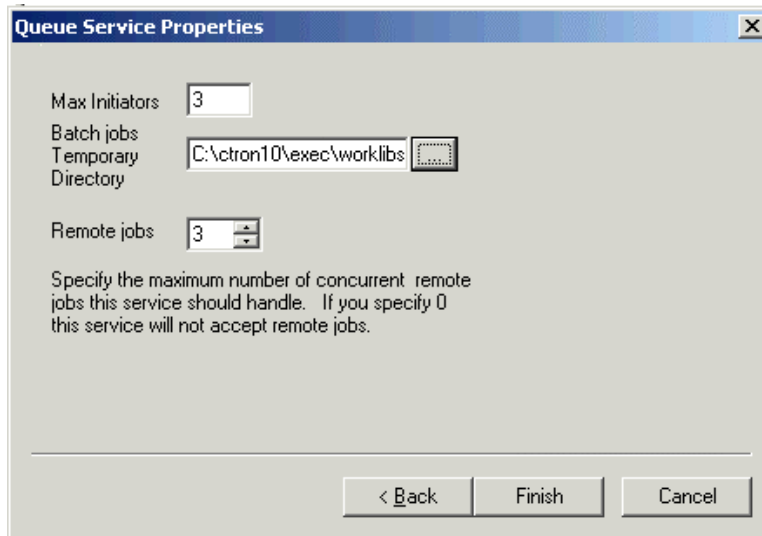
**Configure Initiators Dialog Box**

13. Click OK. The initiators have been configured.
14. Configure the .device file on the application server.
  - ◆ From the tree in the left panel of the Computron Server Administrator, expand (by double clicking) Environment.
  - ◆ Double-click Config files to display its sub-items.
  - ◆ Click .devices to display its contents in the right panel.
  - ◆ To .devices, add an entry similar to this: 1<tab>local<space>C:=dataserver@2000
    - 1 – An ordinal number. First entry should be “1”, second entry “2”, etc.
    - local – A keyword (used for runtime) and must be in lowercase.
    - C: – The drive's letter where Computron is installed on the Application Server.
    - dataserver – The name of the Remote Server.
    - @2000 – The port number used when creating the distributed hold service on the remote server.
  - ◆ On the ConfigFile menu, click Save.
15. Stop and restart the CSServe service.



## *Installation – Remote Server*

1. Share Data with the Application Server. On the remote server:
  - ◆ Map 2 drives to directories on the Application Server:
    - Map one drive (for example D:) to the Application Server's data directory (usually \ctron\data).
    - Map a second drive (for example S:) to the Application Server's spool directory (usually \ctron\var\spool).
2. Configure the LGMAP on the remote server.
  - ◆ From the tree in the left panel of the Computron Server Administrator, expand (by double clicking) Environment.
  - ◆ Double-click Config files to display its sub-items.
  - ◆ Click LGMAP to display its contents in the right panel.
  - ◆ In LGMAP, change the DATA and SPLVOL entries:
    - DATA D:
    - SPLVOL S:
  - ☞ *Note:* If you mapped different drive letters, use them instead of D: and S:
3. On the ConfigFile menu, click Save.
4. Stop and restart the CSServe service.
5. Install the Remote Server following steps 2 to 13 as described earlier in this section. After step 13, the installation for the remote server is complete. The remote server's batch queuing is now listed in the Service Management list.
6. If not using the Copy Configuration Parameters from an existing service feature the Queue Service Properties dialog box is displayed.



**Queue Service Properties dialog box**

Enter the following information:

### **Max Initiators**

Enter the maximum number of Batch Queuing initiators that are configured for the service. This determines the maximum number of jobs that can run at the same time.

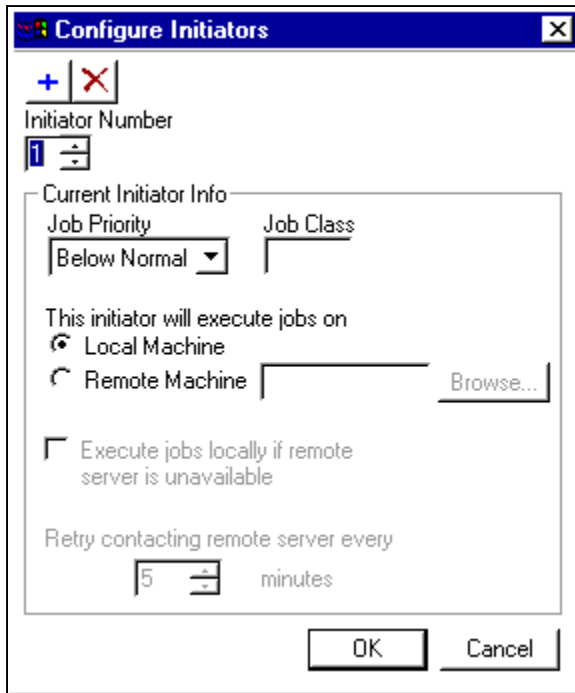
### **Batch jobs Temporary Directory**

Use the browse button to locate the temporary work directory (Joblib). A typical directory location is C:\Ctron\var\worklibs\Joblib.

### **Remote Jobs**

Enter the number of remote jobs that can be accepted concurrently. Usually this is the same as the value in the Max Initiators field.

7. Click Finish.
8. If prompted to create a directory, verify the destination and click Yes to proceed.
9. Configure Initiators.

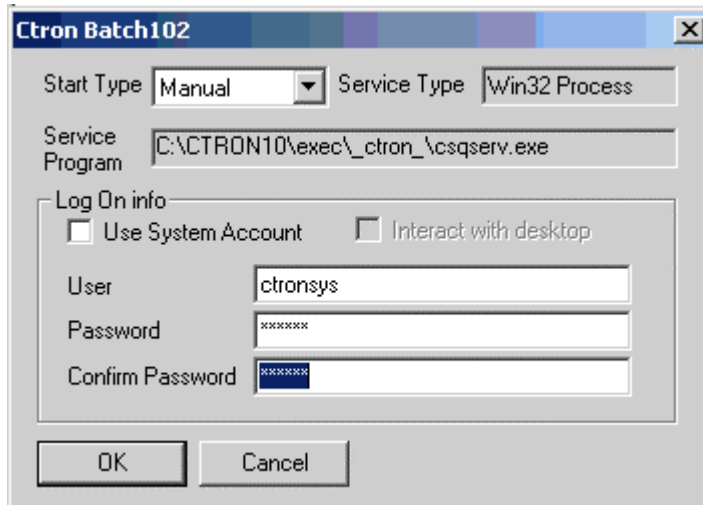


**Configure Initiators Dialog Box**

- ◆ Use the Up/Down arrows next to the Initiator Number field to select the initiator to be configured.
- ◆ Use the default values, click OK.
- ◆ The Initiators have been configured.

### *Remote Server – Final Configuration*

1. Go to “Service Management” in ConfigMan.
2. On the Service menu, select Properties.



Batch Service Priorities for Remote Batch Queuing Server

### Use System Account

Clear this field.

### User

### Password

### Confirm Password

Type ctronsys as the user and supply the password in both password boxes.

3. Click OK.
4. Install a Distributed Hold service. For installation instructions refer to section 5.9, “Installing a Hold Service.”
5. Stop and restart the CSServe service, start the distributed hold service and the Batch Queuing service.

## Batch Queuing Service Maintenance

The Computron Server Administrator enables you to easily maintain the Batch Queuing Service. For example, you can use the Computron Server Administrator to temporarily shut down the Batch Queuing Service if it should require maintenance. This utility also runs the Computron Process Monitor (ListProc), which allows you to view and control both the queued and running batch jobs.

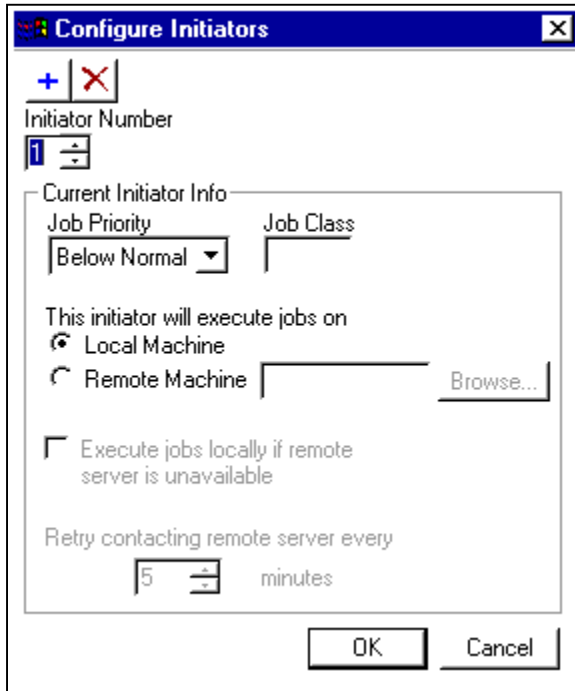
### Adding, Modifying and Deleting Initiators

*Note:* Changes take affect after the service has been stopped and restarted.

1. From the service item tree in the left panel, expand (by double clicking) Services.
2. Double-click the name of the batch service to display its sub-items.
3. Double-click Initiator Config to display the Configure Initiators dialog box.

## Adding Initiators

1. Click Add to add additional initiators.

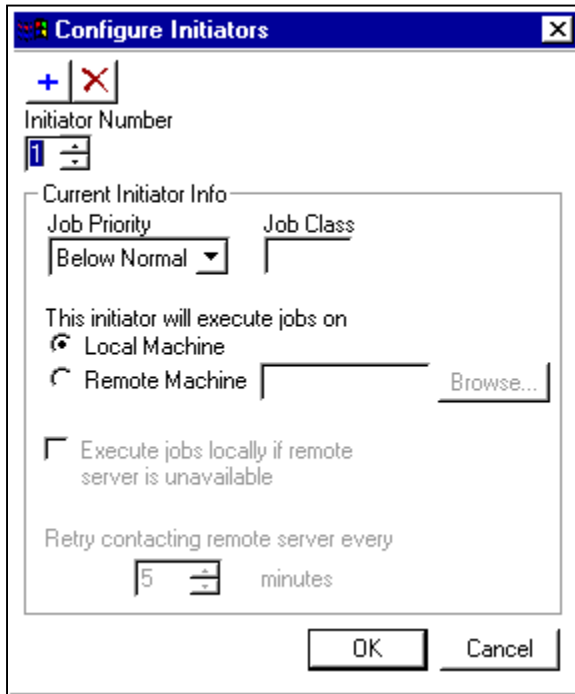


**Configure Initiators dialog box**

2. Complete the information for each initiator. For the definition of the fields on this window, refer to step 12 under the Not Using Copy Configuration Parameters heading at the beginning of this section.
3. To add additional initiators, repeat steps 1 and 2.
4. Click OK.

## *Modify an Initiator*

The information about each initiator can be modified. For the definition of the fields on this window, refer to step 12 under the Not Using Copy Configuration Parameters heading at the beginning of this section.

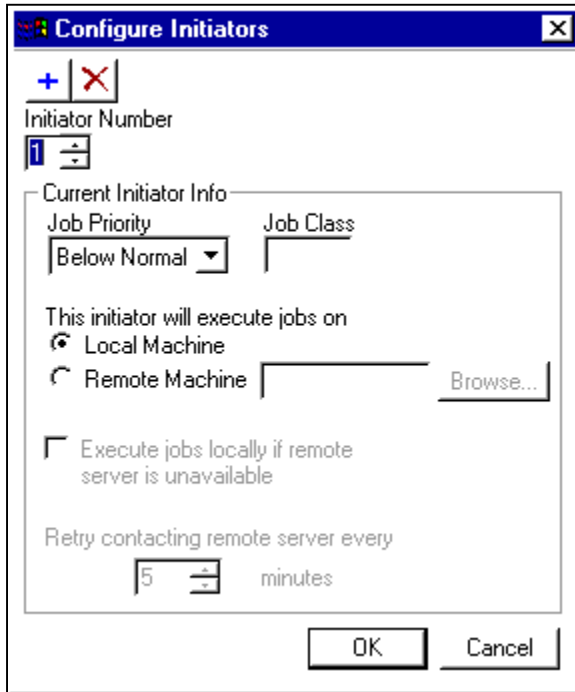


**Configure Initiators Dialog Box**

Click OK. The initiators have been modified.

## Deleting Initiators

1. Use the Up/Down arrows next to the Initiator Number field to select the initiator to be deleted.



**Configure Initiators Dialog Box**

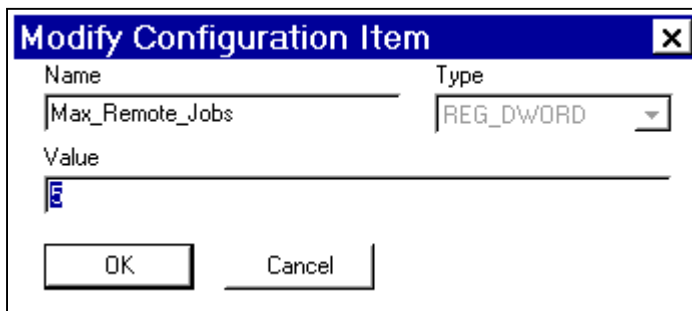
2. Click the Delete button.
3. Click OK to confirm deletion process.
4. The initiator has been deleted.

## Modifying the Number of Remote Jobs



This modification is done on the remote server in a distributed batch configuration.

1. From the service item tree in the left panel, expand (by double clicking) Services.
2. Click the name of the batch service to display configuration items in the right panel. Then click on Settings.
3. Select Max\_Remote\_Jobs.
4. On the toolbar, click Modify. Make the changes to the Value field.



Modify Configuration Item –Max\_Remote\_Jobs dialog box

5. Click OK.
6. Stop and restart the Batch Queuing Service.

## Starting and Stopping Services

### *Starting a stopped Batch Queuing Service*



1. If the services are not displayed in the right panel, click Service Management in the left panel.
2. In the right panel, select the Batch Queuing Service to be started.
3. On the toolbar, click Start.
4. Confirm the Status change from Stopped to Running (from red to green).

 **Warning:** Stopping and restarting the Batch Queuing Service purges all batch jobs waiting or being held in the queue.



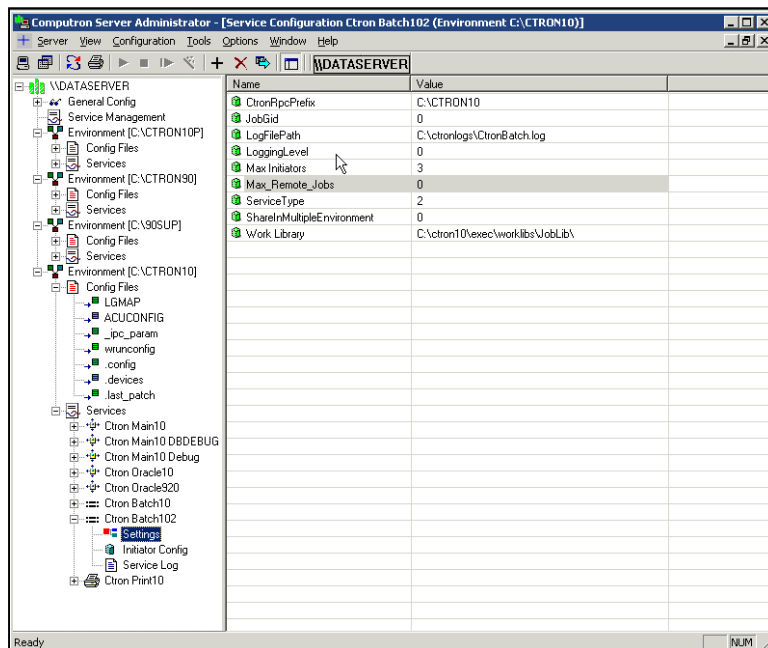
## Stopping a Started Batch Queuing Service

**Important!** Before stopping a service, use the Computron Process Monitor for Windows to verify that the batch queue is empty. For detailed instructions, refer to Section 5.1.1, “Computron Process Monitor,” later in this guide.

1. If the services are not displayed in the right panel, click the Service Management option in the left panel.
2. In the right panel, select Batch Queuing.
3. On the toolbar, click Stop.
4. Click Yes to confirm your decision.
5. Confirm the Status change from Running to Stopped (from green to red).

## Editing Batch Queuing Service

The Computron Server Administrator provides an easy way to maintain these settings, thereby eliminating the need to run the Windows Registry Editor. Keep in mind, however, that many configuration items have default values that should not be modified unless directed to do so by a Computron representative. These items were assigned appropriate values when you ran Setup to initially install the software.



**Batch Service Configuration (Sample) Items**

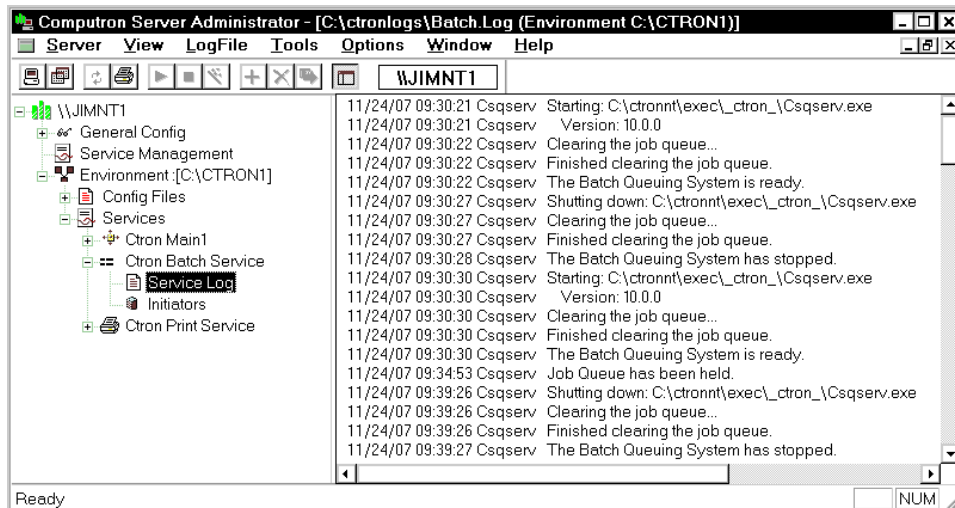
## The Batch Queuing Log File

The Batch Queuing System creates and maintains a log file that contains informational messages for all job queue activity. It also contains various error and warning messages. In many respects it is identical in function to CtronSystem.log, which is the log file for the main Computron connection service, CSServe. The main purpose of the Batch Queuing log file is to provide full accountability of all jobs submitted to the Batch Queuing System.


### Accessing the Batch Queuing Log File

Use the Computron Server Administrator to access the log file for the Batch Queuing service.

1. Expand the Services entry for the Computron environment.
2. Expand the Batch Service entry.
3. Double-click the Service Log entry under the Batch Service.



Batch Queuing Service Log Entries

4. To print the log file, select it and click the Print icon  on the toolbar.

## Configuring the Batch Queuing System

### Setting Run Priorities

Perhaps the easiest part of configuring the Batch Queuing System is determining the run priority settings – there are only two recommended settings from which to choose. However, a little knowledge of how priorities work on Windows is required. There are two basic guidelines to follow. The first is that background jobs must be run at a lower priority than interactive jobs. The second is that all initiators for a given job class should have the same run priority (this guideline can be freely ignored if you understand the implications).

☞ *Tip:* The easiest method for setting your run priorities is to set all initiators to Below Normal. You may, therefore, defer reading this part. It becomes required reading, however, if you ever experience system performance problems. In which case, you may need to adjust priorities to Lowest.

### *Computron vs. Windows Priorities*

First, let's take a quick look at how Windows handles priorities. The following information about priority classes is informational. The important concept to focus on is the priority settings within a priority class (which is explained after this brief introduction to Windows priority classes).

Windows supports the following process priority classes (please note that Windows priority classes and Computron job classes are totally unrelated concepts).

- ◆ Real-Time
- ◆ High
- ◆ Normal
- ◆ Idle

All Computron processes run in the Normal priority class. This includes both interactive (foreground) processes and background processes. The other priority classes are not for general use. A process can only reasonably compete for resources within its own priority class or lower. If other non-Computron processes are run in the High or Real-Time priority classes, the performance of Computron processes may suffer. All applications (both Computron and non-Computron) should be run in the Normal priority class. If that is the case, then we can forget about priority class (because everything is running in the same priority class) and focus our attention on the priority setting within the priority class.

It's helpful to think of each Windows priority class as a small range of priorities. To fully specify a priority, you must specify a priority class, as well as an offset within that priority class's range of priorities. Each priority class has the following range:

Offset	Priority Name	Description
+2	Highest	two points above normal for the priority class
+1	Above Normal	one point above normal for the priority class
0	Normal	the normal priority for the priority class – the default
-1	Below Normal	one point below normal for the priority class
-2	Lowest	two points below normal for the priority class

Most interactive Computron processes run at the Normal priority (again, within the Normal priority class). Batch oriented Computron processes that are run interactively run at Below Normal (and can be configured to run at Normal, Below Normal, or Lowest).

The following table compares the allowable and recommended priorities for all types of Computron processes. The important point to remember is that priorities are relative. A given job is always running at a priority that is equal to, higher than or lower than other jobs it is competing for resources with. When comparing priorities, it does not matter how much higher or lower one priority is, it only matters that it is higher or lower.

☞ *Note:* A column that is conspicuously missing from the following table is one that lists the priorities of the non-Computron processes. You must always consider the processing characteristics of the entire system when choosing performance-tuning characteristics such as priority. In other words, consider the relative priorities of all processes—not just Computron processes.

Priority Name	Process submitted to background (to Batch Queuing System)	Batch-oriented process run interactively (report, update, etc.)	Interactive process
Highest	Not allowed	Not allowed	Not allowed
Above Normal	Not recommended	Not allowed	Not allowed
Normal	Not recommended	Not recommended	Default (only)
Below Normal	Recommended (default)	Recommended (default)	Not allowed
Lowest	Recommended	Recommended	Not allowed

As you can see from the table, interactive processes run at Normal. This means that background processes must run at Below Normal or Lowest. The default configuration causes all batch-oriented processes to run at Below Normal.

### *Possible Configuration Scenarios*

One possible configuration is to set the main initiators (the ones with a blank job class) to Lowest. Then create one or more additional initiators with a non-blank job class (such as A). These additional initiators could be set to Below Normal (one point higher than the main initiators). All jobs would be submitted with a blank job class and would therefore queue up waiting for the main initiators. If a particular job needed to be run right away, its job class could be changed to A, which would cause it to be immediately run by the higher-priority initiator.

Another possible configuration is one aimed at encouraging users to submit batch-oriented jobs, rather than running them interactively. For this scenario, configure the initiators to run Below Normal. Configure batch-oriented processes that are run interactively to run at Lowest. Tell your users that their jobs run faster if they are submitted this way. This is a true statement assuming that the user's job did not get queued behind other jobs. However, since managing the workload increases system throughput, the user's job indeed completes sooner.

Ordinarily, all initiators of a particular job class are given the same priority. This was stated above as the second guideline for setting priorities. However, with an understanding of how initiators select jobs, yet another possibility exists. It may be beneficial to configure a pool of initiators. Essentially, you must consider the system-wide workload and determine which types of processes should have a higher or

lower priority than other types. Consider the relationships between background jobs (processes run by the Batch Queuing System), regular interactive processes, batch-oriented processes that are run interactively and non-Computron processes.

## Determining the Amount of Initiators to Configure

Typically, a small number of initiators should be created. Start out by configuring initiators so that three or four background jobs can run simultaneously. Monitor the performance of the background processes, as well as the interactive processes. Then, adjust the number of initiators accordingly.

### *Temporary Job Files*

Each queued or running background job has an associated job file. A job file is a very small file (typically about 5K), and contains the following information about a job:

- ◆ Name and location of the program to run;
- ◆ Environment information (needed to construct the Computron environment).

The job file contains all the information needed to run the job except for user security information. Additionally, there is a small amount of job information in the memory of the service, such as user security information and job queue status information. It is important to understand that both sets of information are needed in order for a job to run. If either the job file or the in-memory information are lost, the job(s) cannot be run. There are implications that need to be understood (such as what happens to queued jobs when the service is stopped).

The Computron Batch Queuing System requires that you create a directory to store job files. Create a directory called Joblib in the worklibs directory (usually `c:\ctron\exec\worklibs`). The name of this directory is fully configurable and the directory may be placed anywhere. However, you should follow the Computron conventions whenever possible as it simplifies things when speaking to a Computron support representative. This directory never grows large (it is cleaned out, if necessary, when the service starts). Lastly, this directory does not need to be backed up or recovered in the event of a system failure (although doing so provides better tracking of job status at the time of the system failure).

## 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.

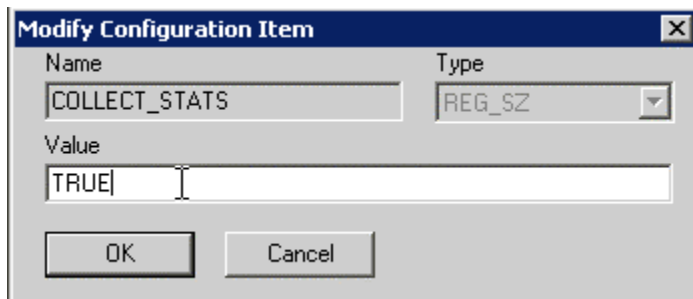
### *Setup for the Analyzer*

1. To collect statistical data for the Analyzer, the following entry must be made in the desired CSServe service using the Computron Server Administrator. For more information on adding an entry to a CSServe service, see Section 5.2 “Computron Server Administrator.”

Name: COLLECT\_STATS

Type: REG\_SZ

Value: TRUE



**Modify Configuration Item for CSServe service**

2. Stop and restart the CSServe service.

*Note:* For information on restarting a CSServe service, see Section 5.2, “Computron Server Administrator.”

The raw data is stored in a file called CSMasterStat.Dat located in the `.../var/Stat` directory. Without this entry, 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. Run NTStart on the Windows server.
2. Execute the following command on the Windows server 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"
```

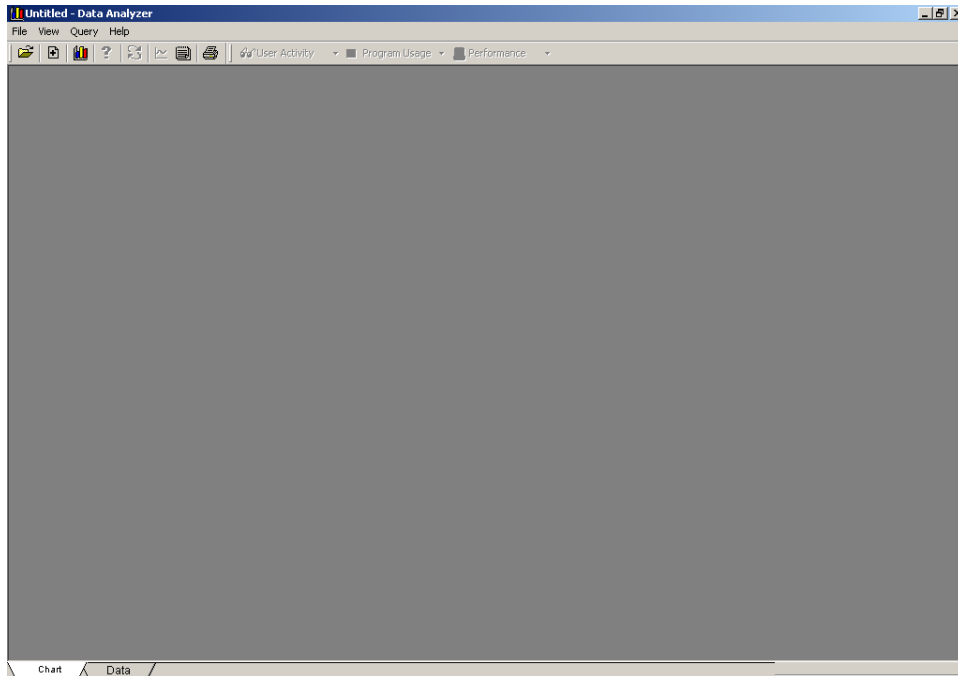
☞ *Note:* The CollectStat program cleans out the original .dat file. That means that any information that was in the original .dat file is now only available in the .mdb file it was converted to. CollectStat does not delete the CSMasterStat.dat file, so it starts collecting data again.

☞ *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.

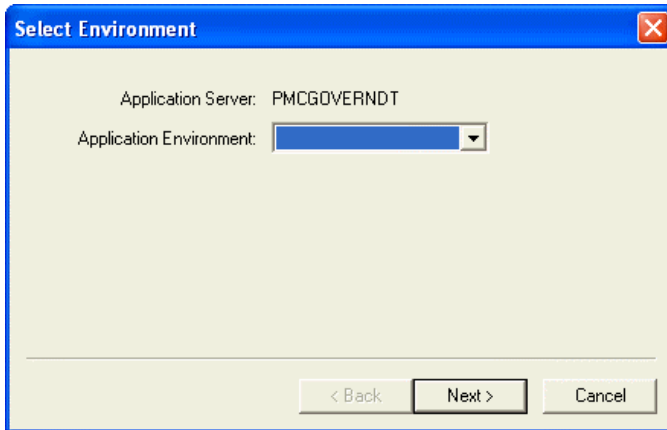
☞ *Note:* The Access database file is the raw data file, converted in a previous step.

☞ *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*

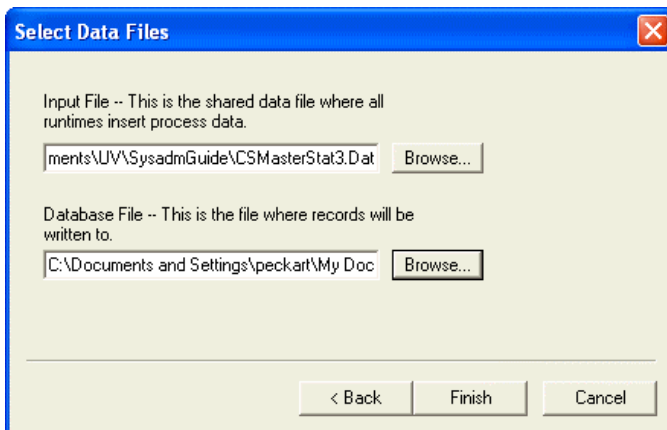
1. Select Import New Data from the File menu, to import data from the local Computron application server.
2. Select the environment where the Analyzer data was collected. You can leave this field blank.





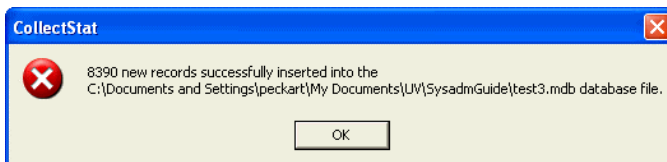
**Import Data Select Environment Window**

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.

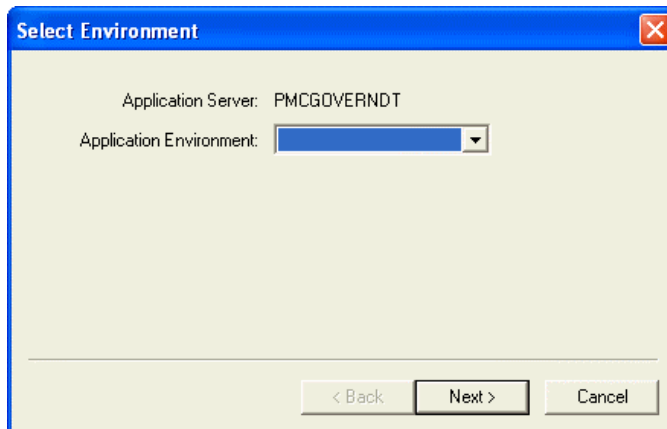
*Note:* This process runs the CollectStat program which cleans out the original .dat file. That means that any information that was in the original .dat file is now only available

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.

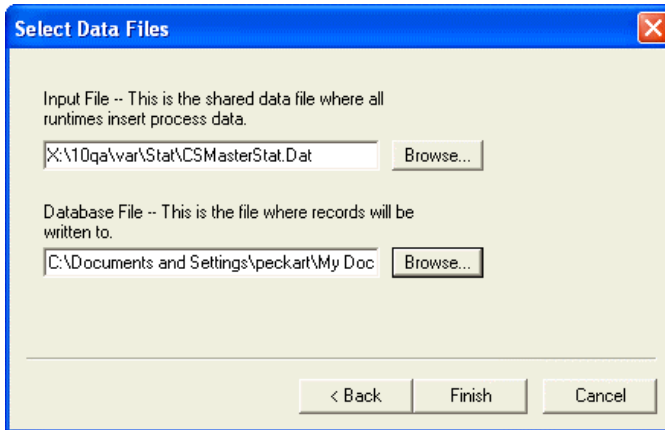
## Adding data to an already existing database file

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.



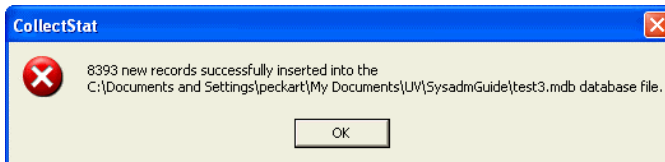
**Import New Data Select Environment Window**

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.



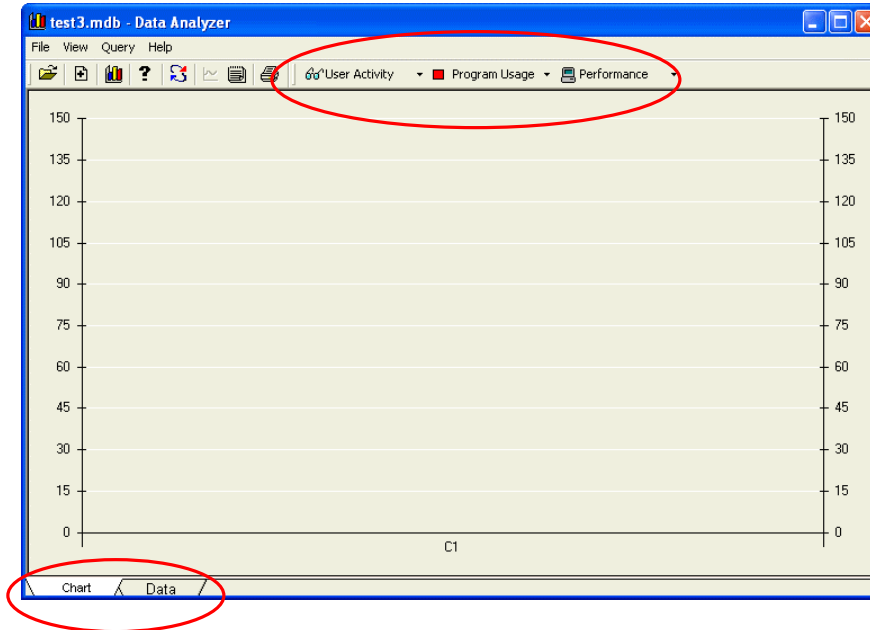
**Import New Data Message when data is added to an existing database**

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

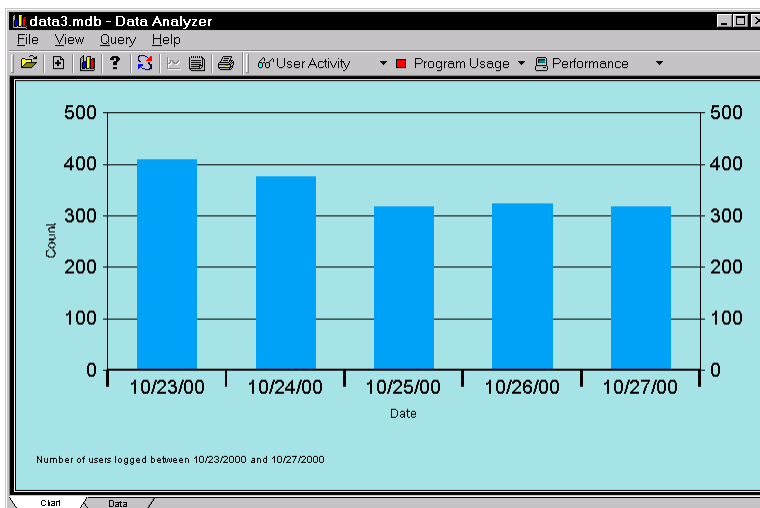
The data is organized and displayed by selecting one of the analysis options from the User Activity, Program Usage or Performance menus.



*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.

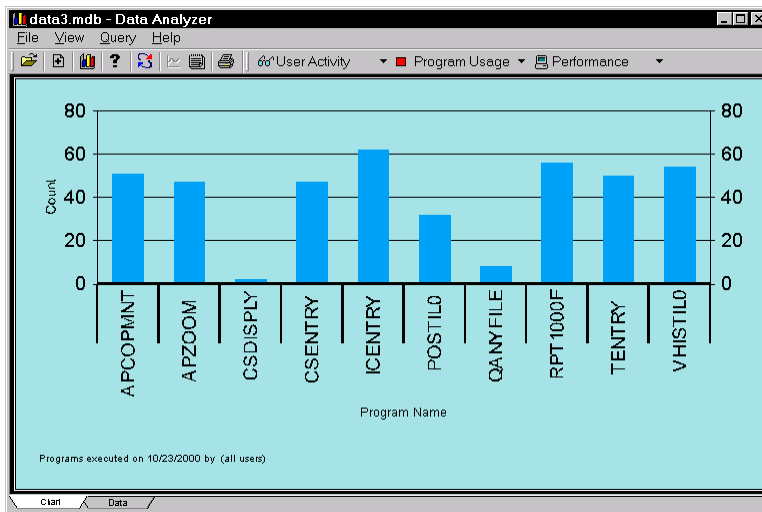
### User Activity Menu:

- ◆ **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.



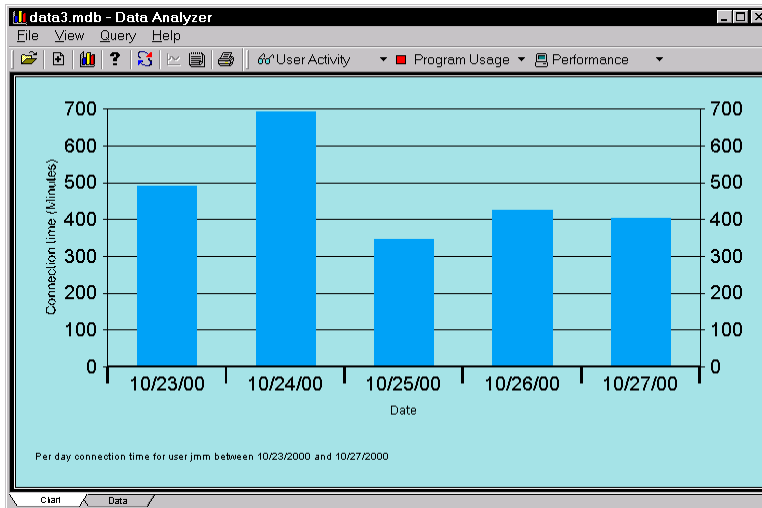
**Total users logged on in a given date range**

- ◆ **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.



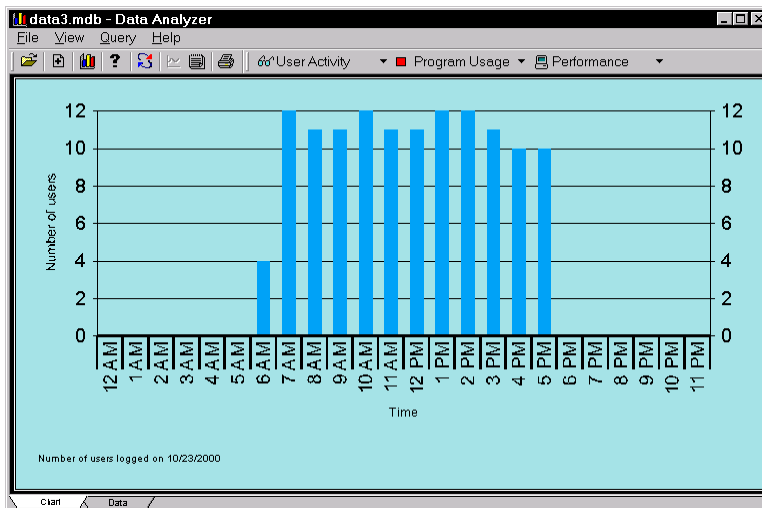
**Programs executed on a particular day**

- ◆ **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.

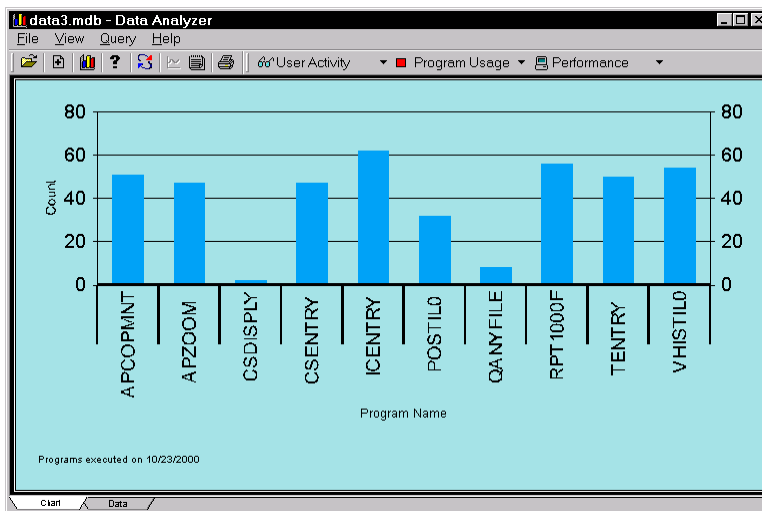
☞ *Note:* There is no “Chart View” associated with this query

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

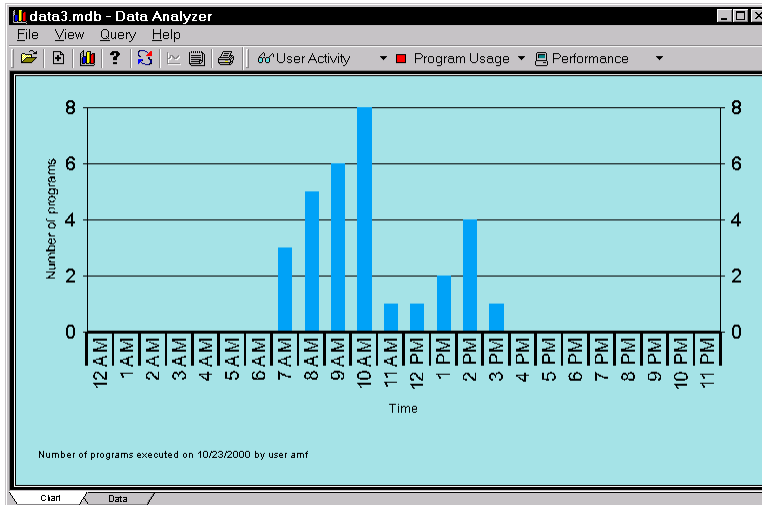
## Program Usage Menu:

- ◆ **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

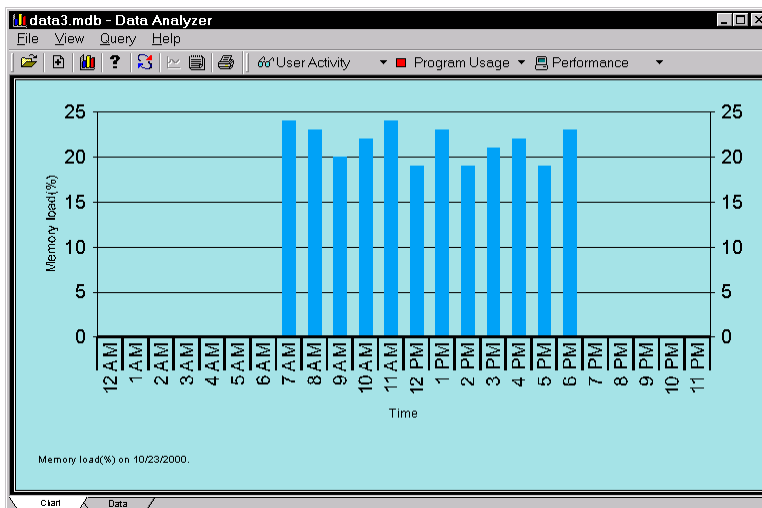
- ◆ **Programs executed vs. time on a particular day** - This output is a timeline (similar to “Number of Users Logged On” above) but it allows filtering by a particular user (shown here) or “all users”. This shows how many programs have been run during the time periods shown.



Programs executed by time

## Performance Menu:

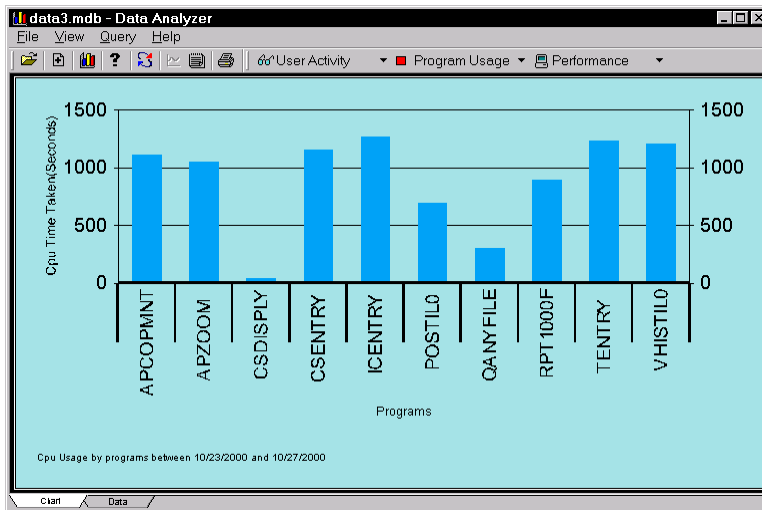
- ◆ **Memory Load** - This tracks the memory load on the system (in percent) over the course of a single day. This can be used to isolate memory usage spikes or to see if in general more memory is needed on the system.



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”.



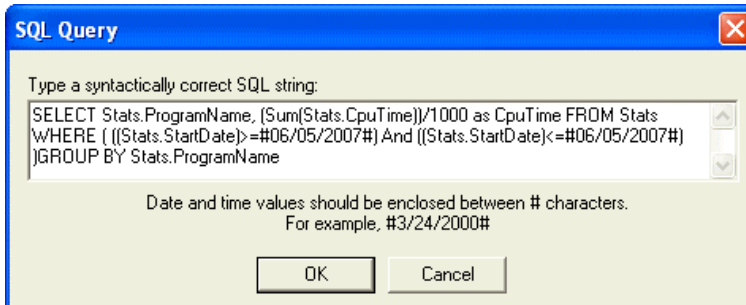
CPU usage by program

## Using SQL Queries:

Choose User SQL Command... from the Query menu.

*Note:* It is possible to formulate a query to get specific data from the Analyzer database.

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.

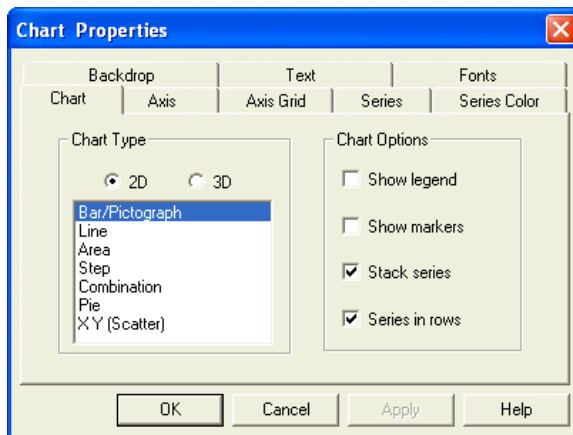
## Viewing Options:

### 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....

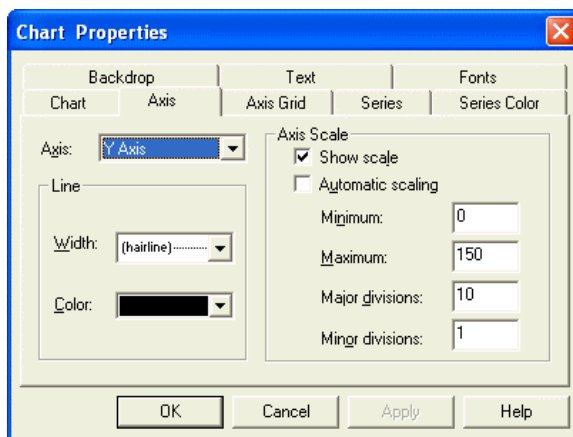
Select the Chart tab to modify a desired property.



**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.

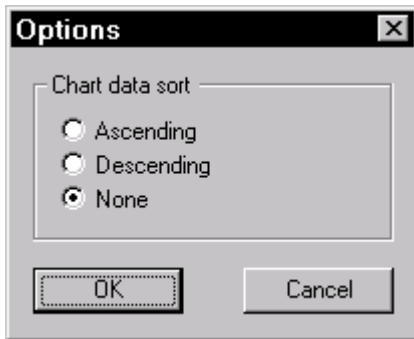
## Options...

Using the Options dialog, the data in the chart can be sorted in ascending or descending order.

From the View menu select Options....

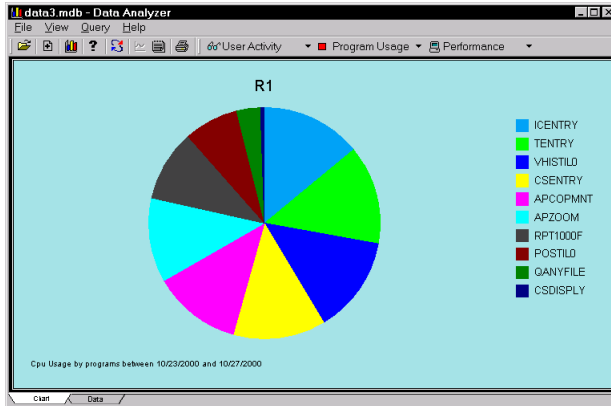
Select the order that the data should be sorted.

*Note:* The ascending/descending sort only applies to certain charts.

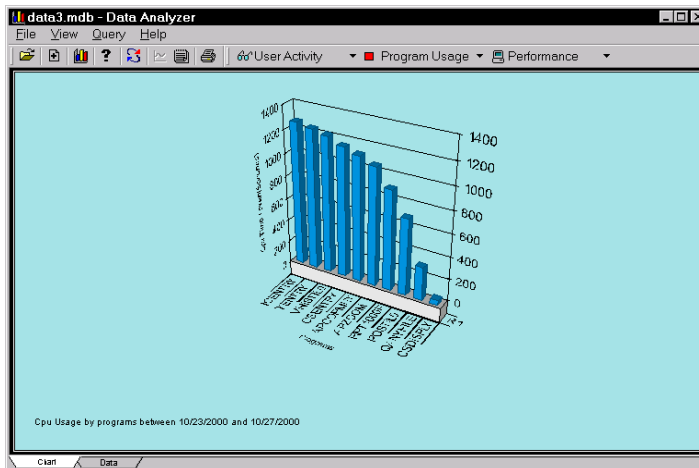


**Chart sorting options**

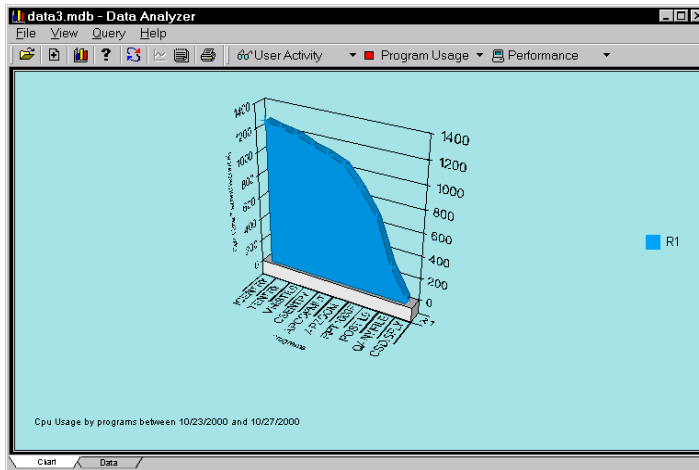
### Example Charts



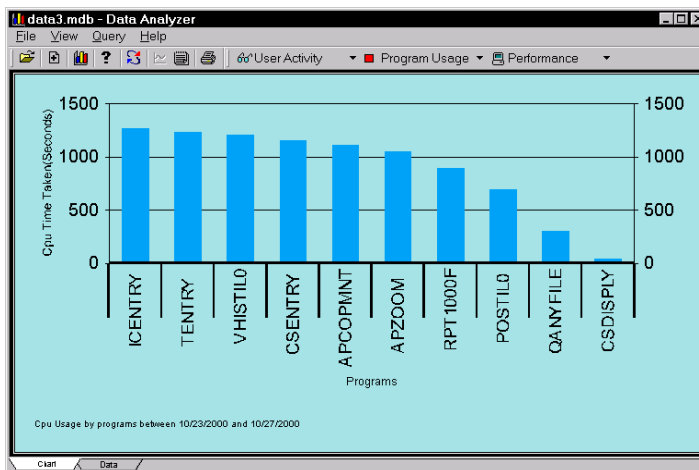
**Pie Chart**



**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,EPICWF ,19089,19089,0,770,0,4/13/2007,2:58:25 PM,4/13/2007,4:50:34 PM
pse,FORMINIT,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,AUDUVPRO,22760,22777,0,59,0,4/16/2007,8:48:49 AM,4/16/2007,8:48:57 AM
pse,FORMINIT,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,AUDUVPRO,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,AUDUVPRO,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,AUDUVPRO,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,AUDUVPRO,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,DBMAMNT,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


```

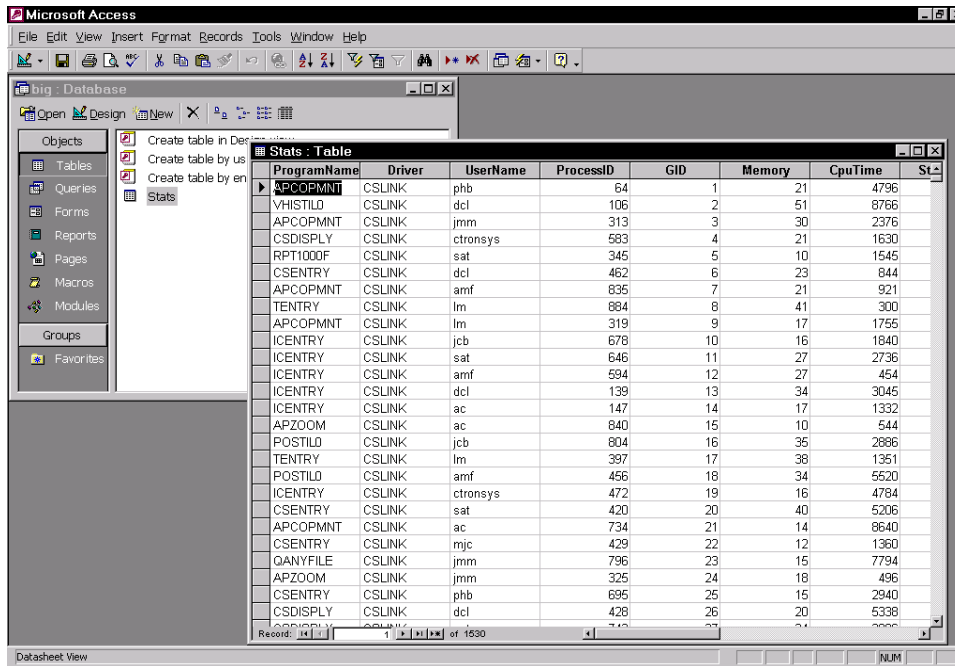
<?xml version="1.0" encoding="UTF-8"?>
<Elements>
<Element id="30" User="pse" GID="22760" PID="22760">
  <Program>FORMINIT</Program>
  <Driver>CSHDVRK</Driver>
  <Memory>0</Memory>
  <CpuTime>29</CpuTime>
  <StartDate>4/16/2007</StartDate>
  <StartTime>8:48:16 AM</StartTime>
  <EndDate>4/16/2007</EndDate>
  <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>AUDUVPRO</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|POSTIL0|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
POSTIL0	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](http://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 to Computron's Configuration File*

The behavior of Computron applications software is influenced by several factors: the Computron runtime, application global environment parameters, Windows service configuration items 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.
- ◆ The Windows environment of the user dictates the Windows configuration items. Among other settings, these configuration items specify values like directory names, TCP/IP port numbers and logging levels.
- ◆ 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 parameter is not applicable to a Windows environment.

#### **COMMAND\_KEY\_PGM**

This parameter is not applicable to a Windows environment.

## 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 is 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 Windows 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 is not applicable to a Windows environment.

## **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 recordsize 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:

csenq – 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: "csenq" and
# "lp"
# CSBATCH_CMD=csenq
CSBATCH_CMD=csenq

#DB_ENABLE_CONCATENATED_SELECTS=NO
BULKLOAD_ENABLE=YES
```

## 5.5 Creating Scripts via Computron's Procedure Generator

### *Introduction to Computron's Procedure Generator*

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 for Computron's Procedures*

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.

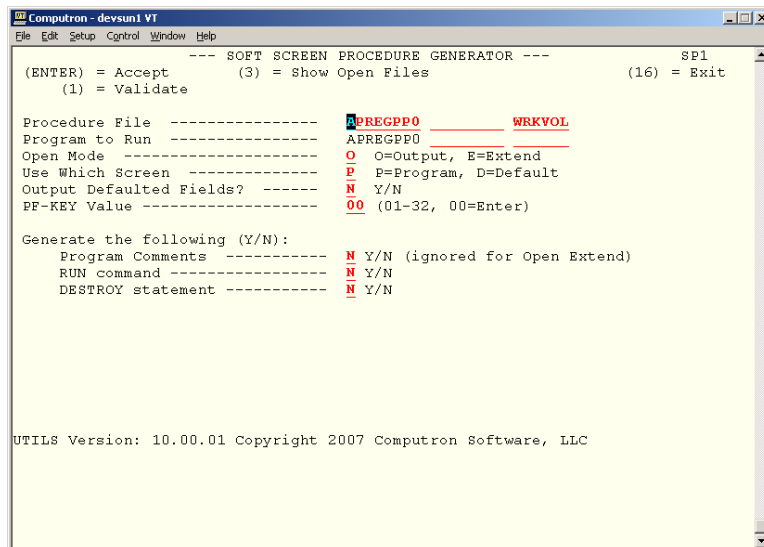
☞ *Note:* Unlike an error message, an information-only message does not cause a window to be displayed. These types of messages can be differentiated from each other in one of two ways:

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).

## Procedure Generator Window (SP1)

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 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 Comptron 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.

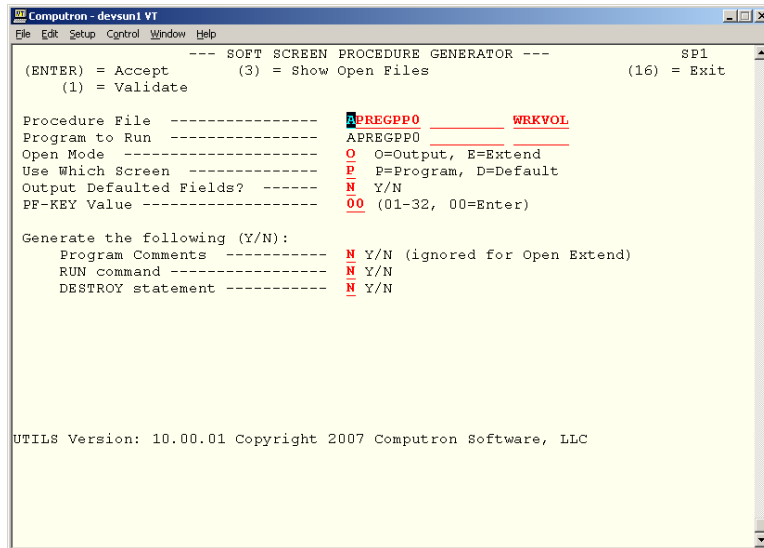
1. Start a Computron character cell session.
2. From its application menu, select the function to be proceduralized. If you know that a front-end program is run from an existing procedure that you would like to include in your customized procedure; i.e. providing values for the GP0 screen, then run the 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.

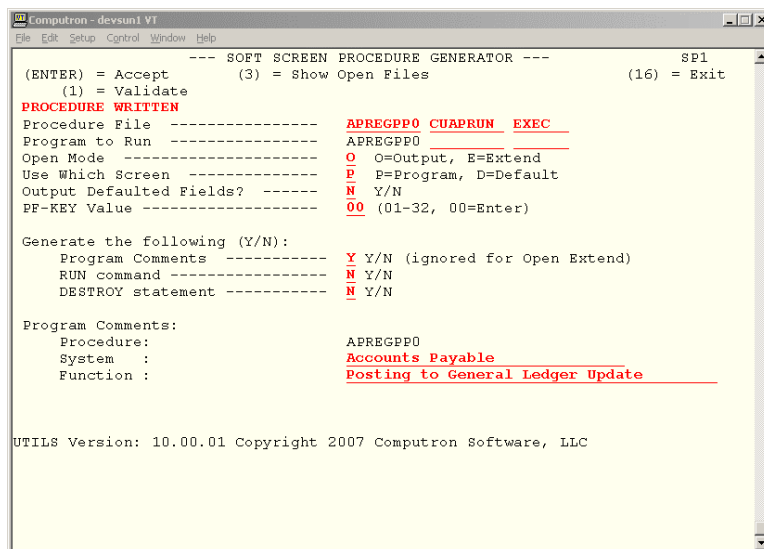
4. Enter data on the window to be written to the procedure.
5. Press PF-31. The current window is displayed in Set Defaults (D) mode. If you enter data on this window, be sure to set the Use Which Screen field to D on the SP1 window.
6. Enter defaults, if any, and press PF-30 to display the Soft Screen 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 field 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. While the function is open, use the Computron Process Monitor to identify the function's Group ID (GID)

☞ *Note:* You may have to adjust your process sort sequence to include the GID option. For detailed instructions, see Section 5.3, Computron Process Monitor.”

4. Run the NTStart utility and from the command prompt, type the following command:

```
csputparm ishow -w GID
```

(where GID represents the Group ID of the function being proceduralized.)

5. The `csputparm ishow` 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.

- Use the copy and paste commands to insert the PROGLIB and PASSMSG1 lines into your procedure immediately after the cspuparm ENTER GTUSEROV line, as follows:

```

cspuparm ENTER GTUSEROV \
PROGLIB  =CIAPRUN
PASSMSG1 =CIAPRUN      S   AP   000   REPORT                N   "  \

```

- 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 CA-UNICENTER, or Windows scheduler. The following sections describe the steps required to access the Job Scheduler from a Visual Basic 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*

Computron’s NTStart utility initializes the environment of the logged in user and enables you to run Computron scripts. The actions performed by NTStart and some of the variables set by NTStart are dependent upon the arguments specified on the command line. Specify a command line causes NTStart to run your scripts within the NT console session.

To accomplish this, you must edit your procedures so that NTStart loads the environment variables for the Computron environment and so that it expands them appropriately to run your procedures (e.g., `myscript`) within the MKS Korn shell. For example, you must add the following lines at the beginning of every procedure:

```
"n:/.../exec/_ctron_ntstart.exe -c "Ctron Script"  
"n:/mksnt/sh.exe n:/.../exec/cuuvload/myscript""
```

☞ *Note:* For more information on using the NTStart utility, refer to Section 4.2, “Computron’s NTStart Utility,” 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 ignored by NTStart since they only pertain to UNIX implementations of Computron software.

```

# 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 next series of lines tell NTStart to load the environment variables for the Computron environment and to expand them appropriately to run your procedures (e.g., myscript) within the MKS Korn shell.

```

"n:/.../exec/_ctron_ntstart.exe -c "Ctron Script"
"n:/mksnt/sh.exe n:/.../exec/cuuvload/myscript"

```

The ENTER CSPHSET 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 CSPHSET 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"

```

*Note:* This “csputparm” is one continuous line. There are no carriage returns until the end. Adding carriage returns prevents the information form being correctly executed.

```

csputparm ENTER KS2 PFKEY=00 MISCCOD1=" " MISCCOD2=" " MISCCOD3=" " APPLYIND=" "
APPPFRDTC=" " 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 Comptron software environment.

```
"n:/.../exec/_ctrn_ntstart.exe -c "Ctron Live Script"  
"n:/mksnt/sh.exe n:/.../exec/cuuvload/myscript"  
  
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=" "
```

 *Note:* the / at the end of each line is a continuation character. When this line is executed, the script interpreter understands that the next line is a continuation of the previous line.

```
csputparm ENTER GTUSEROV PROGVOL=EXEC  
wrun bsjobpp2  
CTRON_PROG_RETURNCODE=`wretcode`  
export CTRON_PROG_RETURNCODE  
csputparm clear
```

## 5.6 Application Server Directory and File Structure

### *Introduction to Server Directories and Files*

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 Window shell scripts, 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 `csuvlipc`.

#### **\$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:

- ◆ 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 Windows Processes: A Technical Overview

Computron applications control and manage several types of processes on Windows operating systems. These processes can be divided into two types: system and user. Both processes are created automatically upon log on, and then are reused as often as possible during application processing for efficient utilization of Windows resources.

Many different Computron processes can be managed from within the Computron Server Administrator, which enables you to configure, monitor, and manage the Computron application servers running under Windows, and the Computron Process Monitor, which enables you to manage the active Computron processes for all Computron users.

☞ *Note:* Refer to Section 5.2, “Computron Server Administrator,” and Section 5.11 “Computron Process Monitor” for detailed instructions on using these tools.

### *System Processes*

Two system processes are created to help administer the Computron software running on Windows:

- ◆ CSServe – this connection service listens for connection requests from Computron clients and creates application processes as necessary.
- ◆ CSMaid – manages the creation, access and removal of various Computron objects as required.

### *CSServe*

A Computron instance is defined as a listening service (CSServe), with a specific set of attributes. This service is controlled by the Computron Server Administrator utility, which can start and stop a service and define its attributes. These attributes are stored in the Windows registry and define such things as the port that CSServe listens to. They can also point to unique Computron installations on the same server in order to provide, for example, access to test and production environments.

Once a service is started, it listens for connection requests from Computron clients: AUI requests on the main port or Telnet requests on the Telnet port. These requests include the user ID and password (which can be optionally encrypted). The service validates this login information against the Windows security system. If the login information is correct, then the service creates an application process for the user, with the privileges of the user.

The service also sends a message to the CSMaid process, indicating the process ID of the newly created process (see the following section for a description of CSMaid). After the application process is created, it inherits the TCP/IP socket created by CSServe. CSServe then returns to sleep mode and waits for additional connection requests.

## CSMaid

On any Computron application server running under Windows, the CSMaid process is always running. CSMaid is automatically started by the first instance of any connection service and is responsible for managing various Computron objects required by the application processes. These objects are created by CSMaid when the application process is created by the listener. They are then deleted when CSMaid gets a signal that the application process has terminated.

When CSMaid receives a signal from any listener, it creates two objects. The first is a work directory, which is private to the application process and contains information that should be shared across this process and its subprocesses, as required. At the same time, CSMaid creates a shared memory segment for this process, which is shared in a similar fashion by the application process and its subprocesses.

The name of the work directory and the name of the memory-mapped file for shared memory is based on the process ID of the application process. The work directory is currently named wk##### (where ##### is the process ID of the application process).

When CSMaid receives a signal that the application process has terminated, it removes the objects created for that process.

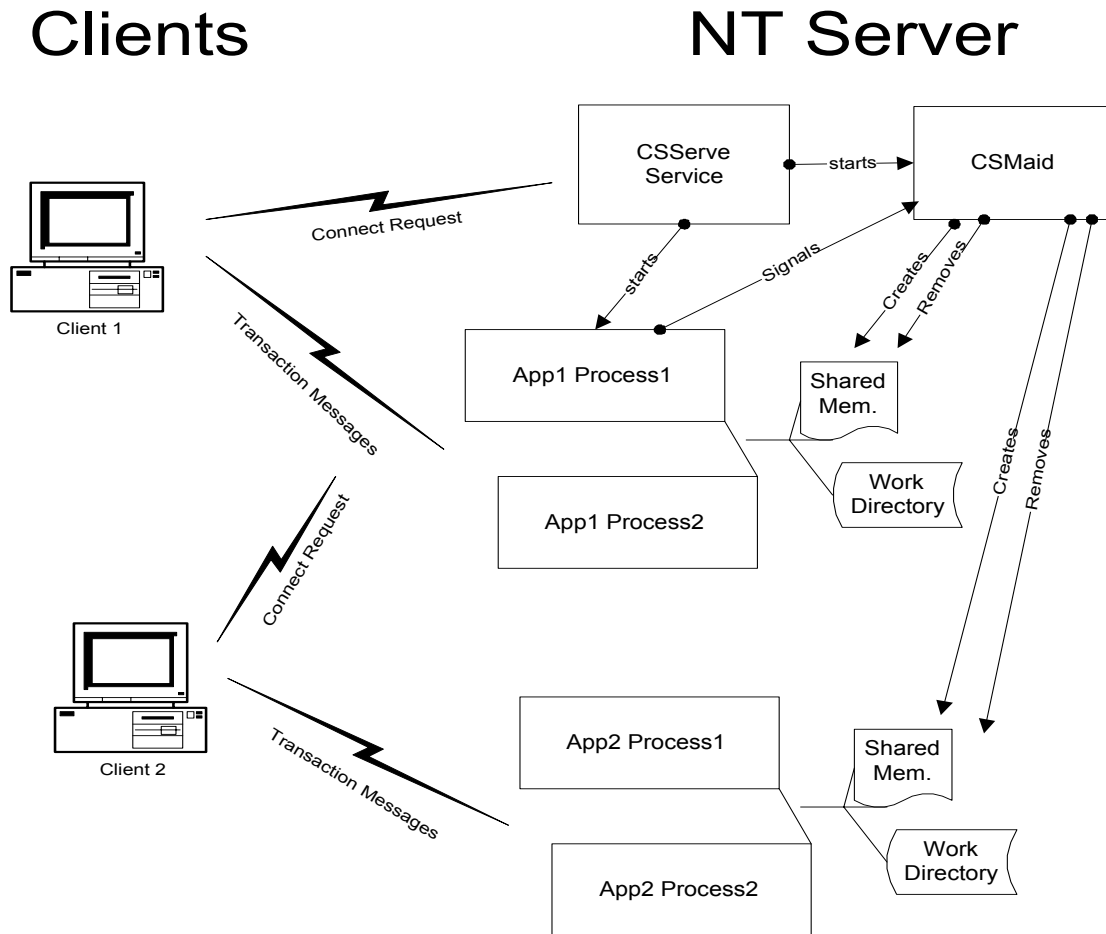
## User Processes

User processes are created by the listener as a result of successful connection requests. One process is created for each connection, as well as one for each client. Normally, two processes are created for each Computron user. The first process's job is to service the Computron Desktop. After initial log on, this process does little actual work.

The other process runs the application portion of Computron's three-tier architecture. Each application a user selects is run in this process. If a user wants to run two applications simultaneously (e.g., Zoom Processing and Transaction Entry), then an additional process is created for that user.

Some applications create an additional temporary subprocess. This process executes additional application steps and is often referred to as the back-end process. For example, a typical Computron application solicits operational information (such as selection criteria) in the front-end process, and then passes that information to the back-end process, which actually generates the report.

This diagram illustrates the flow of the client/server processes on the Windows operating system.





# 5.8 Configuring Computron Print Service

## *Introduction to Computron Print Service*

The Computron Print Service for Windows enables you to handle server-based printing for Computron Financials and Workflow. It also allows you to create customized print forms. The enhanced Print Service functionality is available as of the 4.0.7 or 5.0.1 patch. Previous versions must use the Print Engine (PE32.exe) functionality.

This section includes the following topics:

- ◆ Instructions for installing and configuring an Computron Print Service.
- ◆ A discussion on how to maintain the Print Service.

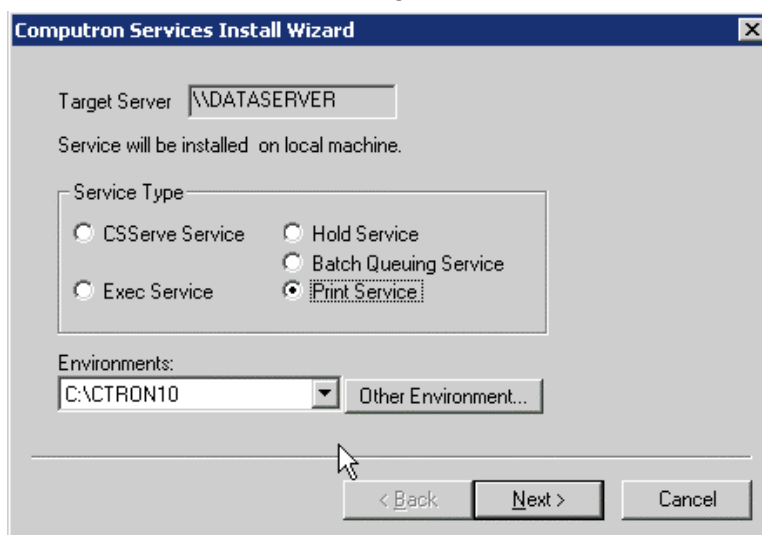
## *Installing the Print Service*

The Computron Server Administrator features a Services Install Wizard that enables you to establish a Computron Print Service. The following instructions explain how to install and configure your service.

The Server Administrator also allows you to install additional Print Services. To make installation of additional services easy, it provides an option to use an existing configured Print Service as a template. This is useful when creating diagnostic versions of services where only a few parameters need to be changed.

1. Open the Computron Server Administrator.
2. From the toolbar, click Install Service.

*Note:* The Service Management list must be active in order to access the Service menu.



**Computron Services Install Wizard dialog box**

Enter the following information:

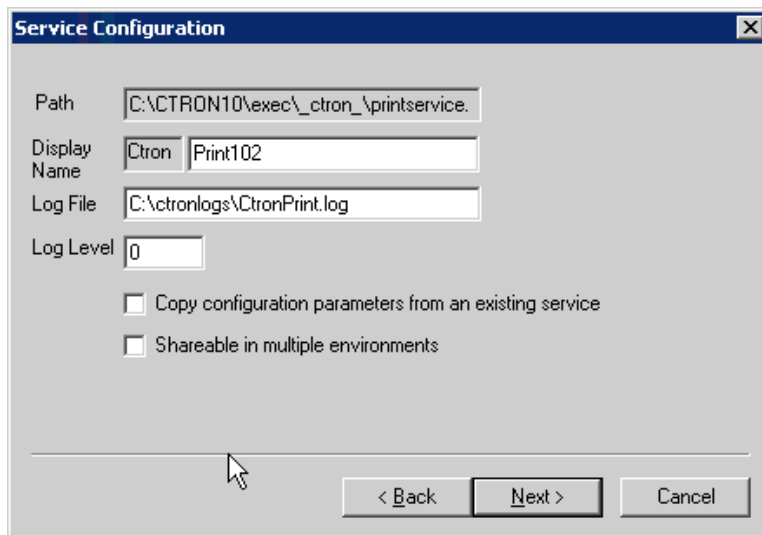
### Choose type of service to install

Set this to Print Service.

### Environments

Enter the environment where you are installing the service. If the environment is not listed, you can click the Other Environment button to browse to the location of the cssrvctl.cfg configuration file for that environment. (This would happen only if there were no configured services in a particular environment.)

3. Click Next.
4. The Service Configuration dialog box appears.



Service Configuration dialog box

Enter service configuration information in the following fields:

### Path

This non-modifiable field displays the path to the executable file for the Print Service. (If the path doesn't match the entry you selected in the Environments box on the previous window, double-check the cssrvctl.cfg file to make sure the CTRON\_ROOT entry is correct.)

### Display Name

Enter a name for the print service.

### Log File

This displays the default file name and location for the service.

### Log Level

This defaults to 0. (Valid values are 0–5, where 0 is normal logging; 1–5 are increased levels of diagnostic logging.)

### Copy configuration parameters from an existing service

Select this check box if you want to use the configuration parameters from an existing Print Service type to create a new service. This option is useful when creating diagnostic versions of services where only a few parameters need to be changed.

### Shareable in multiple environments

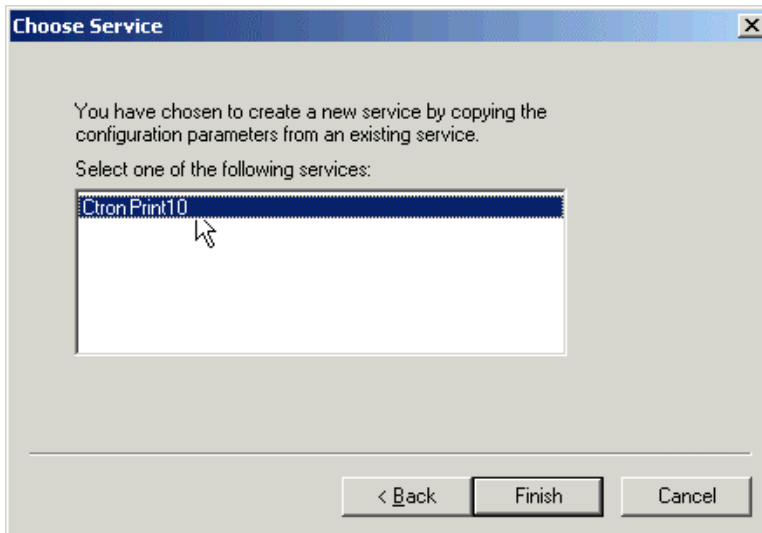
This check box should remain cleared unless otherwise directed by Computron support personnel.

### *Using Copy Configuration Parameters*

- If you selected the Copy configuration parameters from an existing service check box, the Choose Service dialog box is displayed.

☞ *Note:* If the Copy Configuration Parameters from an existing service field is clear, skip this step and proceed to step 10.

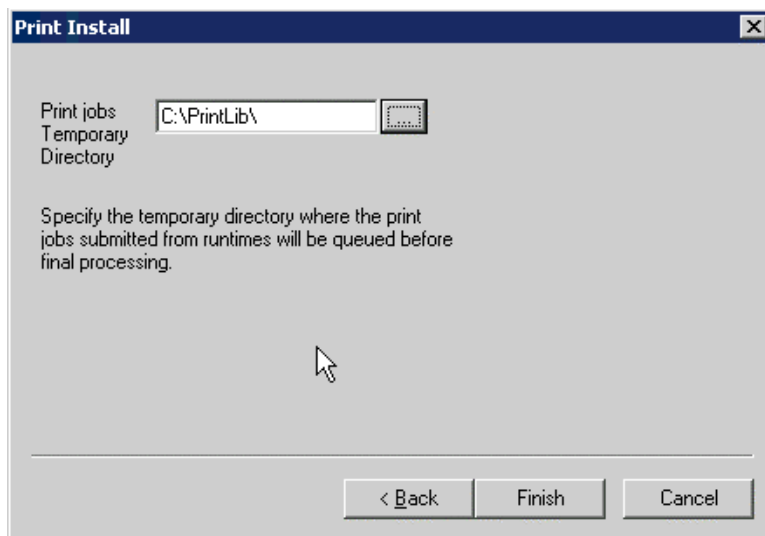
Select the Print Service whose configuration parameters you want to copy.



**Choose Service dialog box**

Click Next.

- On the Print Install dialog box, click on the Browse Button for the Folder Dialog box. Select the location where the system holds the print job. The default location is C:\printlib\.



**Print Install dialog box**

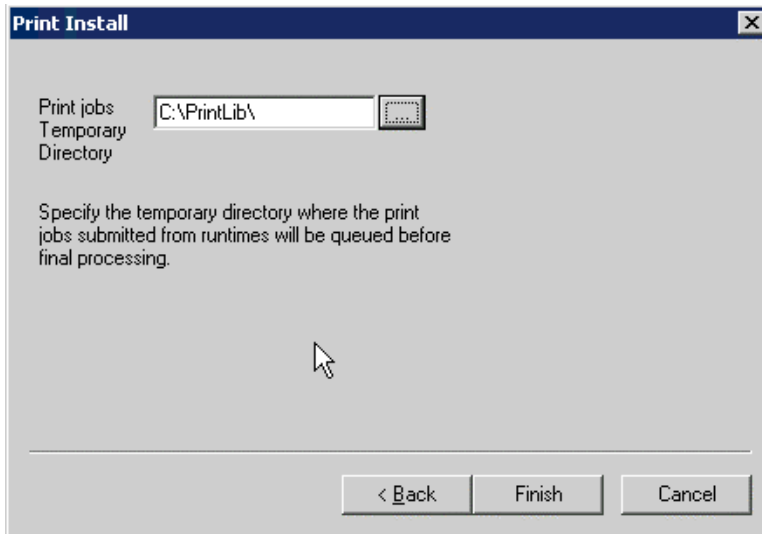
7. Click OK
8. If prompted to 'create directory', verify destination and click Yes to proceed.
9. Click Finish.

You are now returned to the main Server Administrator window. Computron Print Service installation is complete. Go to Print Service Maintenance at the end of this section.

### *Not Using Copy Configuration Parameters*

10. If not using the Copy configuration parameters from an existing service feature, Click Finish (on the Service Configuration dialog box).

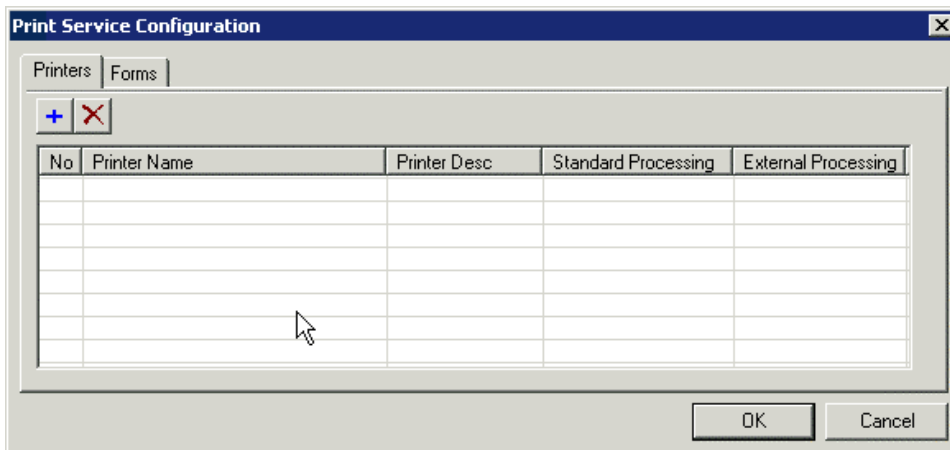
Click on the Browse Button for the Folder Dialog box. Select the location where the system holds the print job. The default location is C:\printlib\.




**Print Install dialog box**

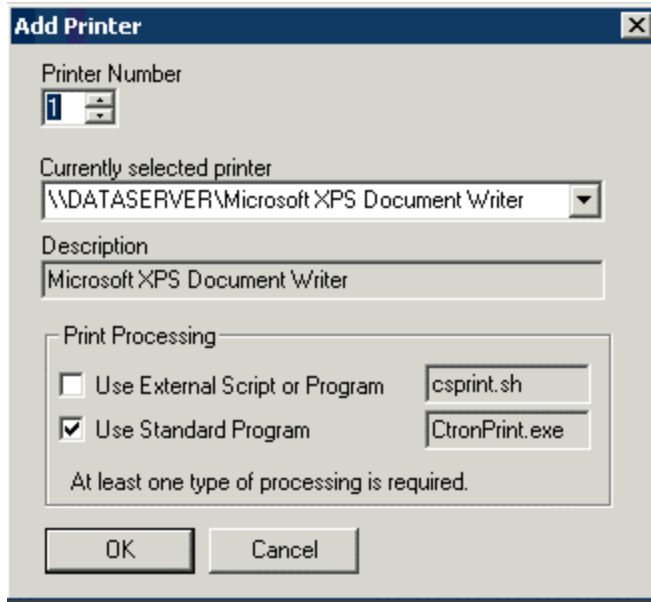
11. Click OK.
12. If prompted to 'create directory', verify destination and click Yes to proceed.
13. From the Print Install dialog box, click Finish. The Print Service Configuration window displays.

You are now ready to create printer definitions.



**Print Service Configuration dialog box**

14. To add a printer definition(s), click .



**Printer Definition dialog box**

Enter the following information:

**Printer Number**

Use the Up/Down arrows to identify the printer to configure.

**Currently selected printer**

Verify that this is the printer you want to configure.

**Description**

This non-modifiable field identifies the make of the selected printer.

*Print Processing Panel*

You must select at least one of these two fields.

**Use External Script or Program**

This identifies what external script or program is run before the standard print request processing. An external program can perform tasks like additional formatting or copying the print file to another location. Valid options are:

- This causes an external shell or executable program to be run prior to the standard print request processing. Although the program name defaults to `csprint.sh`, you can rename it as necessary.
- An external script or program is not run before the standard print request processing. When this field is clear, the Use Standard Program field must be selected.

**Use Standard Program**

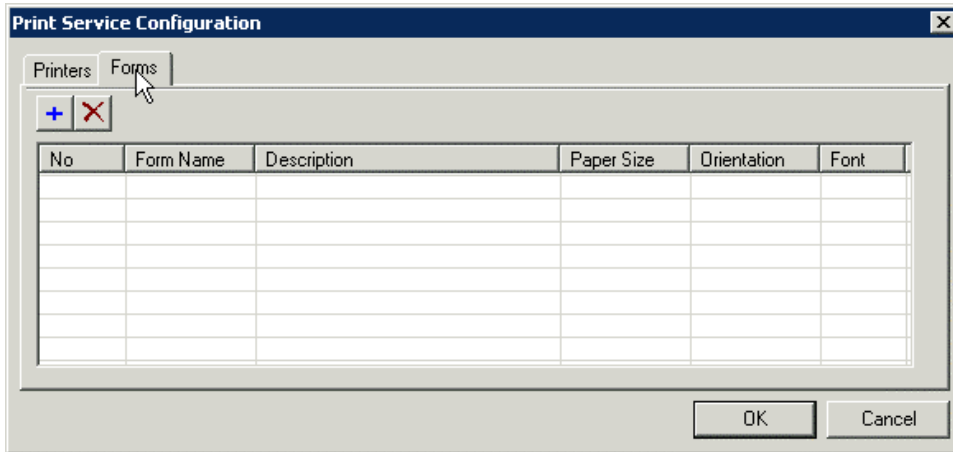
This option determines whether normal printing occurs. Valid options are:

- Normal printing occurs via the program identified in the second field.
- Normal printing does not occur. When this field is clear, the Use External Script or Program field must be selected.


15. Click OK. This saves the definition and returns to the Printer Configuration dialog box.

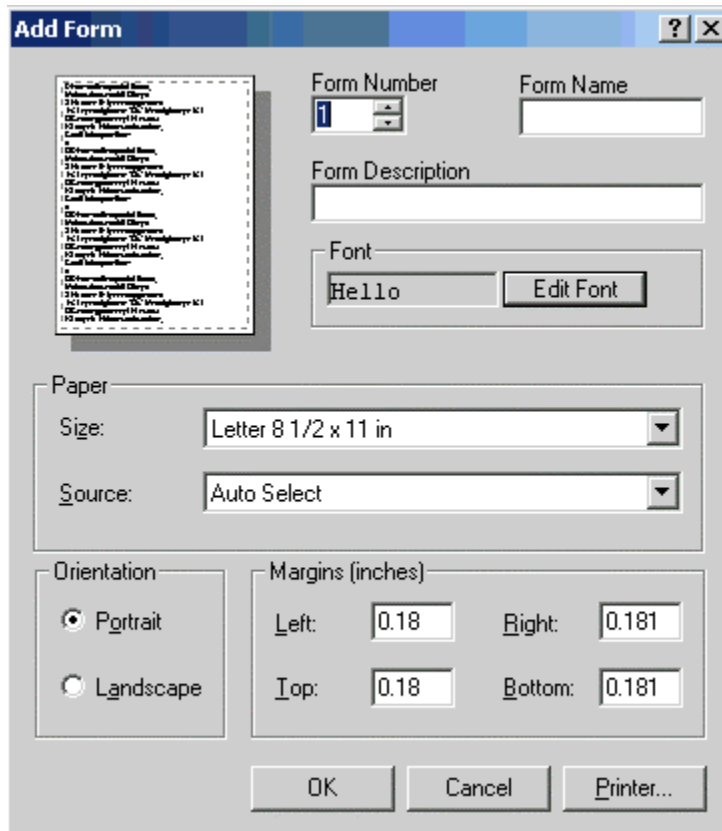
*Note:* To add additional Printer Definitions, repeat step 14 as needed.

16. Now, you must create at least one default form. To create a form(s), click the Forms tab.



**Print Service Configuration dialog box for Forms**

17. To add a form definition, click .



Document Properties dialog box

### Form Number

Use the Up/Down arrows to select the form you want to configure.

### Form Name

Enter a name for the form.

### Form Description

Enter a description for the form.

### Font

Select the Font type. Use the Edit Font button to access more fonts.

*Note:* for work Hello in the Font box previews how the printed output font appears. Although all fonts are available for use when defining a form, we highly recommend using a non-proportional (fixed width) font.

### Size

This is the paper size for printing.

### Source

This is the source of the paper for printing.



## Orientation

This determines whether the output is landscape or portrait.

## Margins (inches)

In these four fields, enter the margins for your printout.

18. Click OK. The form has been saved.

☞ *Note:* To add additional Form Descriptions, repeat steps 17 and 18 as needed.

19. Click OK.

The Print Service has now been added to the Service Management list and the Computron Print Service installation is now complete.

## Print Service Maintenance

The Computron Server Administrator enables you to easily maintain the Print Service. For example, you can use the Computron Server Administrator to temporarily shut down the Print Service if it should require maintenance.

### Starting and Stopping Print Services

#### *Starting a Stopped Print Service*



1. If the services are not displayed in the right panel, click Service Management in the left panel.
2. In the right panel, select the print service that you want to start.
3. On the toolbar, click Start.
4. Confirm the Status change from Stopped to Running (from red to green).

#### *Stopping a Running Print Service*



☞ *Important!* Before stopping a service, use the Computron Process Monitor for Windows to verify that all users are logged off that service. For detailed instructions, refer to Section 5.11, “Computron Process Monitor,” in this chapter.

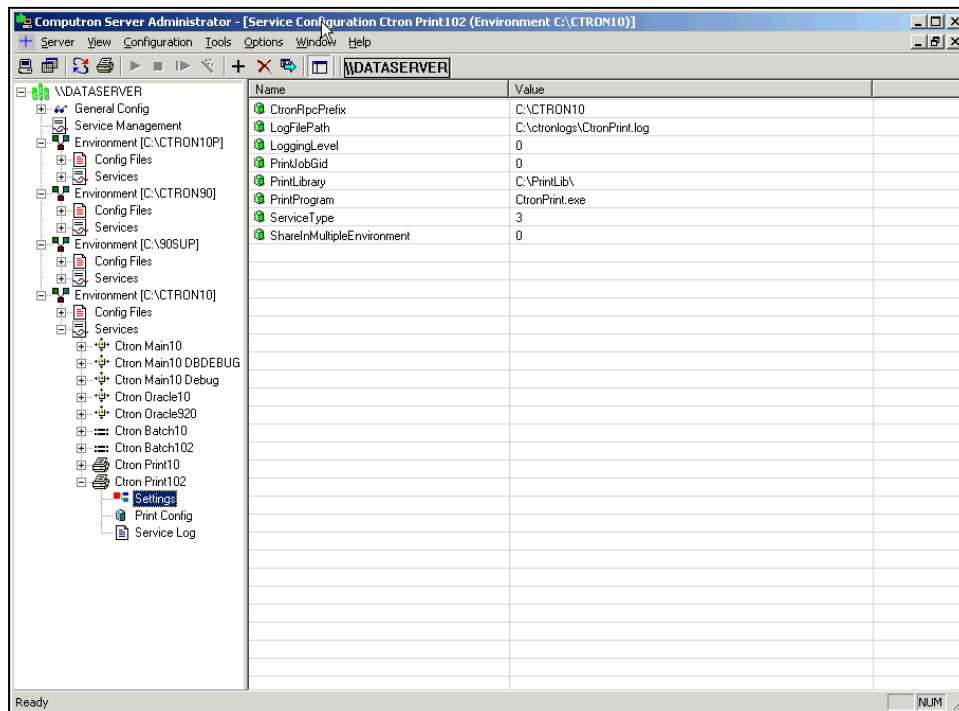
1. If the services are not displayed in the right panel, click Service Management option in the left panel.
2. In the right panel, select the print service that you want to stop.
3. On the toolbar, click Stop.
4. Click Yes to confirm your decision.

5. Confirm the Status change from Running to Stopped (from green to red).

## Modifying the Print Service

The Computron Server Administrator provides an easy way to maintain the Print Service settings, thereby eliminating the need to run the Windows Registry Editor. Keep in mind, however, that many configuration items have default values that should not be modified unless directed to do so by a Computron representative. These items were assigned appropriate values when you ran Setup to initially install the software.

*Note:* See Section 5.1, “Computron Server Administrator,” for detailed instructions on editing a service configuration item.

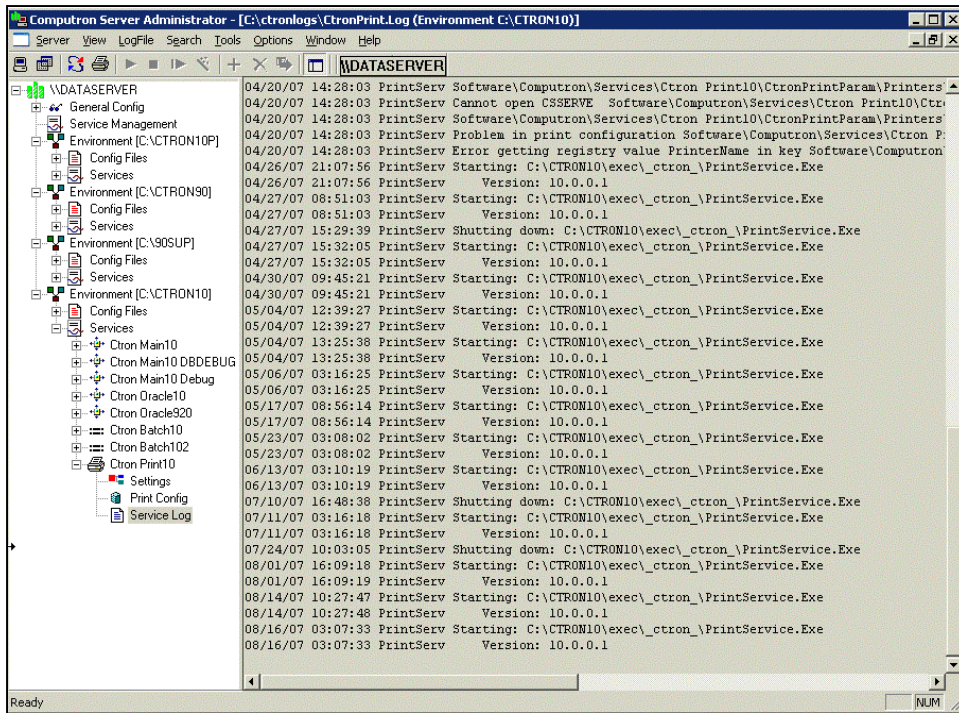


**Print Service Configuration (Sample) Items**


## Accessing the Print Service Log File

The Print Service creates and maintains a log file that contains any error messages that are generated for the print service. Use the Computron Server Administrator to access the log file for the Print Service, as follows:

1. Expand the Services entry for the Computron environment.
2. Expand the Print Service entry.
3. Click the Service Log entry under the Print Service.



Print Service Log Entries

- To print the log file, highlight it and click the Print icon  on the toolbar.

*Note:* Log files are viewable in other formats. See Section 5.1, “Computron Server Administrator,” for detailed instructions on “Editing Log File Viewer.”

## 5.9 Installing a Hold Service

### *Introduction to Installing a Hold Service*

The Computron Hold Service controls the access of data when Computron application programs are running on more than one physical server and are sharing the same database. When two or more application servers are sharing the same database, a Hold Service must be setup on both machines. This allows data access and record locking to be controlled correctly between the application servers.

☞ *Note:* Distributed Background processing is an example of a configuration that requires a Hold Service.

☞ *Note:* Only one Hold service is necessary for each Computron application environment.

⚠ *Warning!* To use the Hold Service, the Computron application must be using a relational database (MSSQL or Oracle). Vision is not supported.

### Installation – Remote Server

1. Run the Computron Server Administrator.

☞ *Note:* For information on using the Computron Server Administrator, see Section 5.1, “Computron Server Administrator.”

2. From the toolbar, click Install Service to display the first Install Wizard dialog box.

☞ *Note:* The Service Management list must be active in order to access the Install Wizard.



**Computron Services Install Wizard dialog box**

Enter the following information in the dialog box:

## Target Server

This non-modifiable field identifies the server.

## Service Type

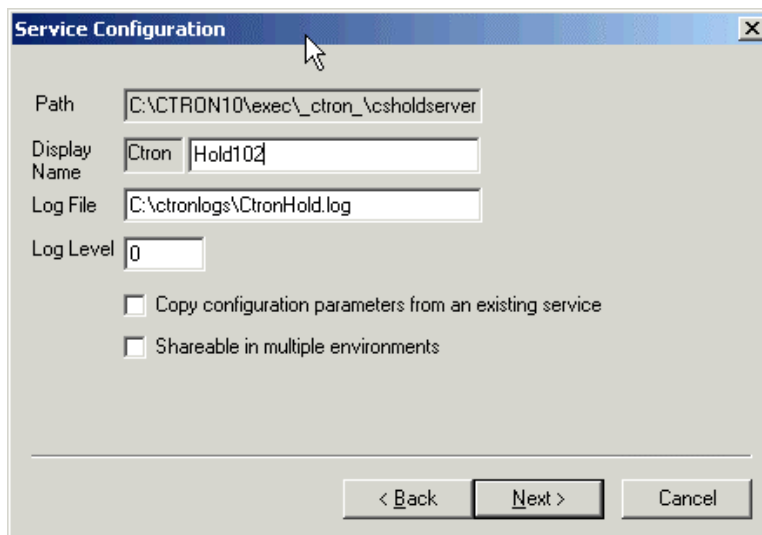
Select the Hold Service option.

## Environments

Select the environment where you are installing the service.

☞ *Note:* If the environment is not listed, click the Other Environment button to browse for the cssrvctl.cfg configuration file for that environment. This would happen only if there were no configured services in a particular environment.

3. Click Next.
4. The Service Configuration box appears.



The image shows a 'Service Configuration' dialog box with the following fields and options:

- Path:** C:\CTRON10\exec\ctron\_\csholdserver
- Display Name:** Ctron Hold102
- Log File:** C:\ctronlogs\CtronHold.log
- Log Level:** 0
- Copy configuration parameters from an existing service
- Shareable in multiple environments

At the bottom, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Service Configuration dialog box

Enter service information in the following fields:

### Path

This non-modifiable field is the path to the executable file for the Hold service. (If the path doesn't match the entry you selected in the Environments box on the previous window, double-check the cssrvctl.cfg file to make sure the CTRON\_ROOT entry is correct.)

### Display Name

Enter a name for the new service.

### Log File

This displays the default file name and location for the service.

## Log Level

This defaults to 0. (Valid values are 0–5, where 0 is normal logging; 1–5 are increased levels of diagnostic logging.)

## Copy configuration parameters from an existing service


Select this check box if you want to use the configuration parameters from an existing Hold Service to create a new service. This option is useful when creating diagnostic versions of services where only a few parameters need to be changed.


## Shareable in multiple environments

This check box should normally be cleared unless otherwise directed by Computron support personnel.

5. Click Next

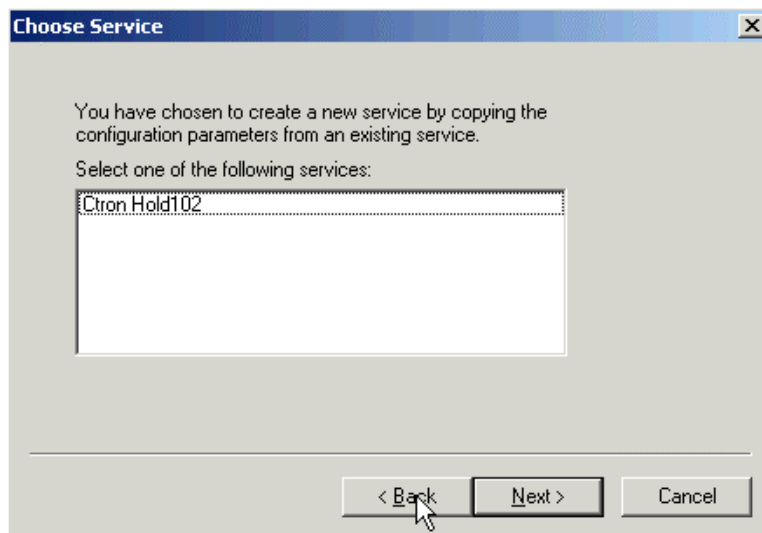
### *Using Copy Configuration Parameters*

 *Note:* If not using the Copy configuration parameters from an existing service feature, skip this step and proceed to step 8.

 *Warning:* Although this feature can be used here, it is recommended that you not use Copy Configuration Parameters.

6. If you selected the Copy configuration parameters from an existing service check box, the Choose Service dialog box is displayed.

Select the Hold Service whose configuration parameters you want to copy.



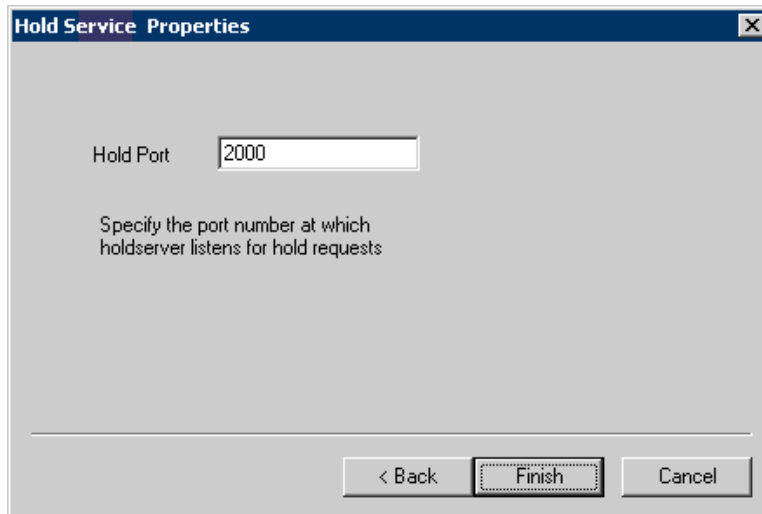
**Choose Service dialog box**

7. Click Finish.

Computron Hold Service, remote server only, installation is complete. The hold service is now listed in the Service Management list.

## *Not Using Copy Configuration Parameters*

8. If not using the Copy Configuration Parameters from an existing service feature the Hold Service Properties dialog box is displayed.



**Hold Service Properties dialog box**

Enter the following information:

### **Hold Port**

Enter the TCP/IP port number that is used by the service.

9. Click Finish.

Computron Hold Service, remote server only, installation is complete. The new hold service is now listed in the Service Management list.

## **Installation – Application Server**

1. Run the Computron Server Administrator to modify the .device file.
  - ☞ *Note:* For information on using the Computron Server Administrator, see Section 5.1, “Computron Server Administrator.”
2. From the service item tree in the left panel, expand (by double clicking) Environment.
3. Double-click Config files to display its sub-items.
4. Click .devices to display its contents in the right panel.
5. To .devices, add an entry similar to this:

```
1<tab>local<space>C:=ctronfin2@2000
```

An ordinal number. First entry should be “1”, second entry “2”, etc.

**local**

A keyword (used for runtime) and must be in lowercase.

**C:**

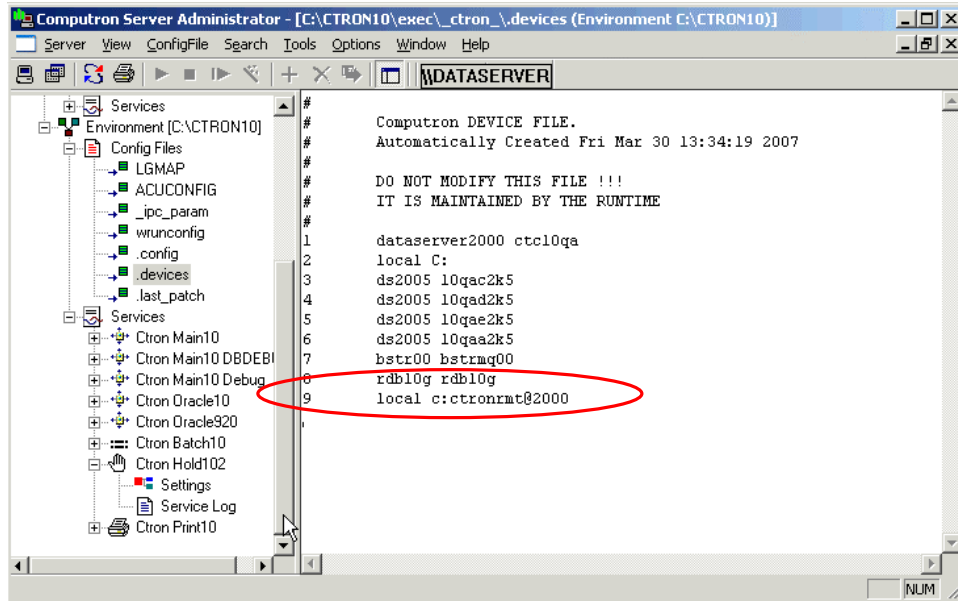
This is the drive's letter where Computron is installed on the Application Server.

**ctronrmt**

This is the name of the Remote Server.

**@2000**

This is the port number used when creating the distributed hold service on the remote server.



**Edit .devices dialog box**

6. On the ConfigFile menu, click Save.



## 5.10 Installing a CSServe Service

### *Introduction to CSServe Service*

The Computron CSServe Service for Windows NT is Computron's main service. The CSServe Service controls connections between the client and the applications. Upon entry of a login request, CSServe validates the username and password, creates work directories where necessary, starts a runtime and registers the runtime with CSMaid. The program then returns to a listening mode and waits for the next request.

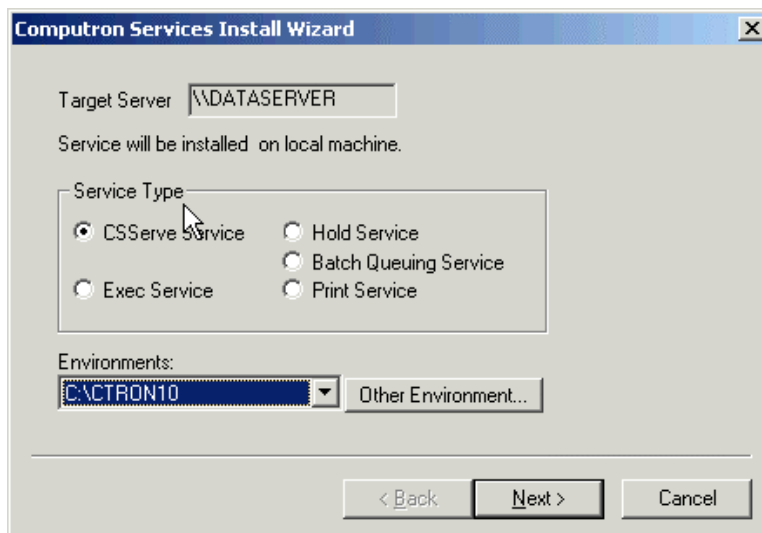
### *Installing a CSServe Service*

The Computron Server Administrator allows you to install additional Computron (CSServe) Services. This is especially useful if you wish to create a backup service for testing purposes. For example, you could have a Production service named CtronProd and a backup service named CtronTest. To make installation of additional services even easier, you have the option to use an existing configured CSServe service as a template.

☞ *Note:* Before proceeding with the installation, it's a good idea to check the cssrvctl.cfg file (in the n:\ctron\exec\\_ctron\_ directory) to make sure the CTRON\_ROOT entry is pointing to the correct environment.

1. From the toolbar, click Install Service to display the first Install Wizard dialog box.

☞ *Note:* The Service Management list must be active in order to access the Install Wizard.



**CSServe – Computron Services Install Wizard dialog box**

Enter the following information:

## Choose type of service to install

Select CSServe Service.

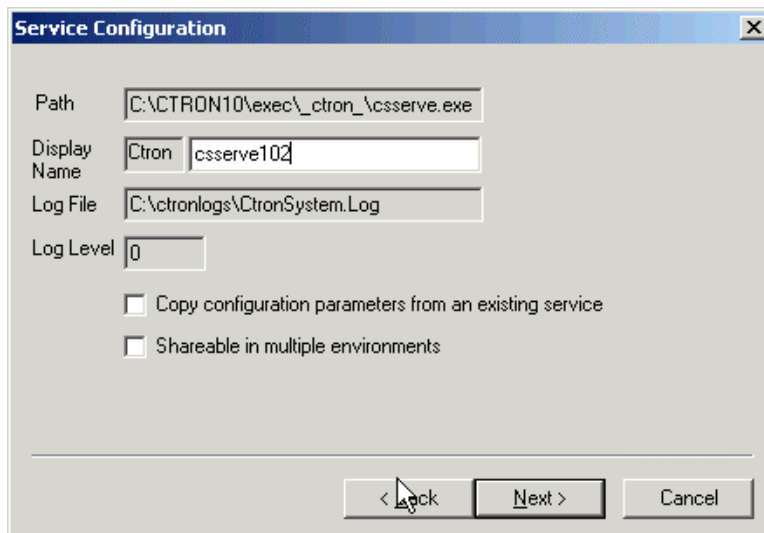
## Environments

Select the environment where you are installing the service.

☞ *Note:* If the environment is not listed, click the Other Environment button to browse for the cssrvctl.cfg configuration file for that environment. This would happen only if there are no configured services in a particular environment.

Click Next.

2. The service Configuration dialog box appears.



**CSServe Service Configuration dialog box**

Enter service information in the following fields:

### Path

This non-modifiable field displays the path to the executable file for the CSServe Service. If the path doesn't match the entry you selected in the Environments box on the previous window, double-check the cssrvctl.cfg file to make sure the CTRON\_ROOT entry is correct.

### Display Name

Enter a name for the CSServe Service.

### Log File

This non-modifiable field displays the default file name and location for the service.

### Log Level

This non-modifiable field defaults to 0. Valid values are 0–5, where 0 is normal logging; 1–5 are increased levels of diagnostic logging. All CSServe Services use the same logging level.

☞ *Note:* Log Level and Log File are only modifiable during the initial installation of CSServe.

### Copy configuration parameters from an existing service

Select this check box if you want to use the configuration parameters from an existing CSServe service type to create a new service. This option is useful when creating diagnostic versions of services where only a few parameters need to be changed.

### Shareable in multiple environments

This check box should remain cleared unless otherwise directed by Computron support personnel.

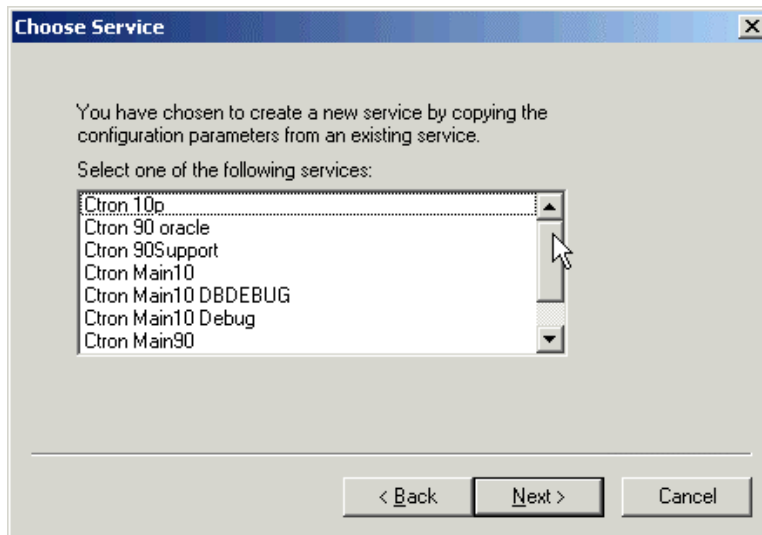
Click Next.

### *Using Copy Configuration Parameters*

*Note:* If not using the Copy Configuration Parameters from an existing service feature, skip this step and proceed to step 4.

- If you selected the Copy configuration parameters from an existing service check box, the Choose Service dialog box is displayed

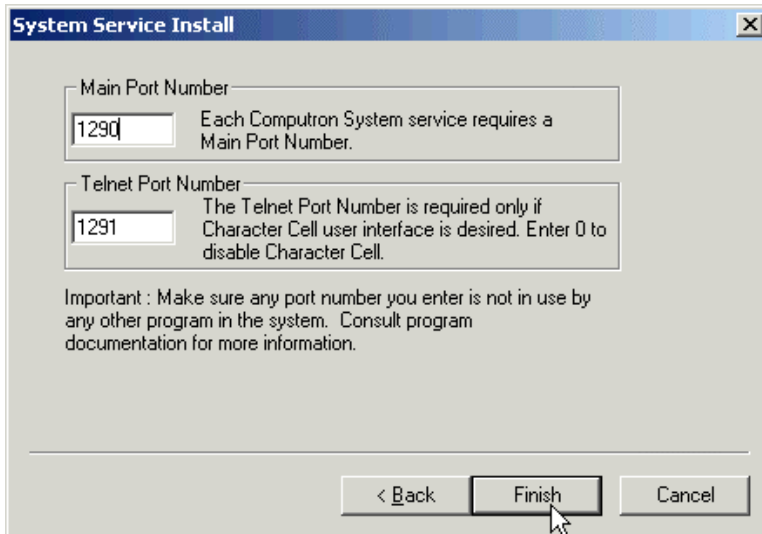
Select the CSServe Service whose configuration parameters you want to copy.



**CSServe Choose Service dialog box.**

Click Next.

In the System Service Install dialog box, enter the following information:



CSServe System Service Install dialog box

### Main Port Number

Enter a valid port number for AUI implementations that access the service.

### Telnet Port Number

Enter a valid port number for character cell implementations that access the service.

☞ *Important:* Make sure any port number you enter is not in use by another program in the system.

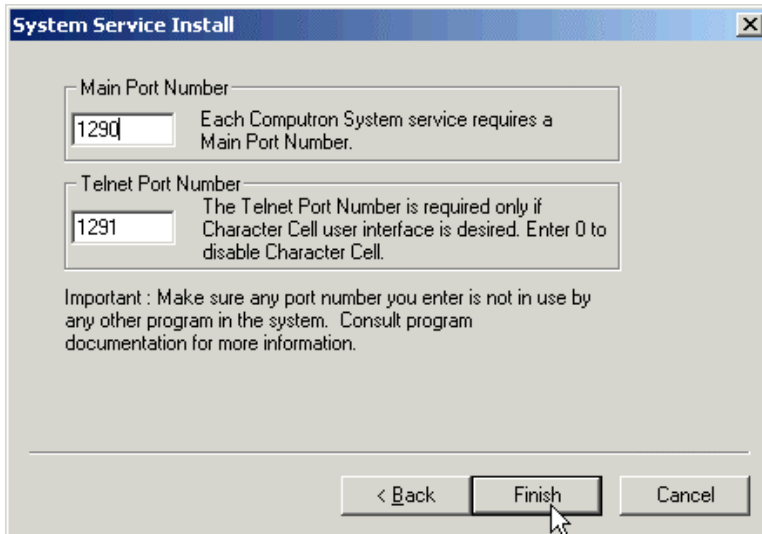
☞ *Note:* If you are using a Telnet port, check the `cssrvctl.cfg` file to ensure that the `CTRON_TELNET_PORT` line exists.

Click Finish.

Computron CSServe Service installation is complete. The new CSServe service is now listed in the Service Management list.

### *Not Using Copy Configuration Parameters*

4. If not using the Copy Configuration Parameters from an existing service feature, the System Service Install dialog box is displayed.



**CSServe System Service Install dialog box**

Enter the following information:

### Main Port Number

Enter a valid port number for AUI implementations that access the service.

### Telnet Port Number

Enter a valid port number for character cell implementations that access the service.

- ☞ *Important:* Make sure any port number you enter is not in use by another program in the system.
- ☞ *Note:* If you are using a Telnet port, check the `cssrvctl.cfg` file to ensure that the `CTRON_TELNET_PORT` line exists.

Click Finish.

Computron CSServe Service installation is complete. The new CSServe service is now listed in the Service Management list.

## CSServe Service Maintenance

The Computron Server Administrator enables you to easily maintain the CSServe Service. For example, you can use the Computron Server Administrator to temporarily shut down the CSServe Service if it should require maintenance.

### Starting and Stopping Services

#### *Starting a stopped CSServe Service*

1. If the services are not displayed in the right panel, click Service Management in the left panel.

2. In the right panel, select CSServe.
3. On the toolbar, click Start.
4. Confirm the Status change from Stopped to Running (from red to green).

### *Stopping a started CSServe Service*



☞ *Important!* Before stopping a service, use the Computron Process Monitor for Windows to verify that all users are logged off that service. For detailed instructions, refer to Section 5.11, “Computron Process Monitor.”

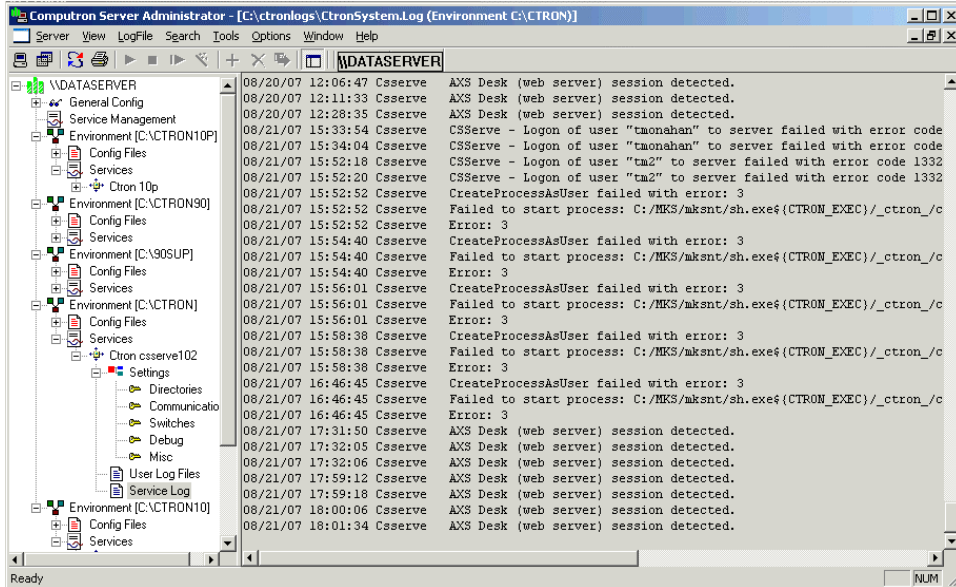
1. If the services are not displayed in the right panel, click the Service Management option in the left panel.
2. In the right panel, select CSServe.
3. On the toolbar, click Stop.
4. Click Yes to confirm your decision.
5. Confirm the Status change from Running to Stopped (from green to red).

## Editing CSServe Service


The Computron Server Administrator provides an easy way to maintain these settings, thereby eliminating the need to run the Windows Registry Editor. Keep in mind, however, that many configuration items have default values that should not be modified unless directed to do so by a Computron representative. These items were assigned appropriate values when you ran Setup to initially install the software.

☞ *Note:* See Section 5.1, “Computron Server Administrator,” for detailed instructions on editing a service configuration item.





#### CSServe Service Log Entries

- To print the log file, select it and click  on the toolbar.

*Note:* Log files are viewable in other formats. See Section 5.1 “Computron Server Administrator,” for detailed instructions on “Editing Log File Viewer.”



## 5.11 Computron Process Monitor

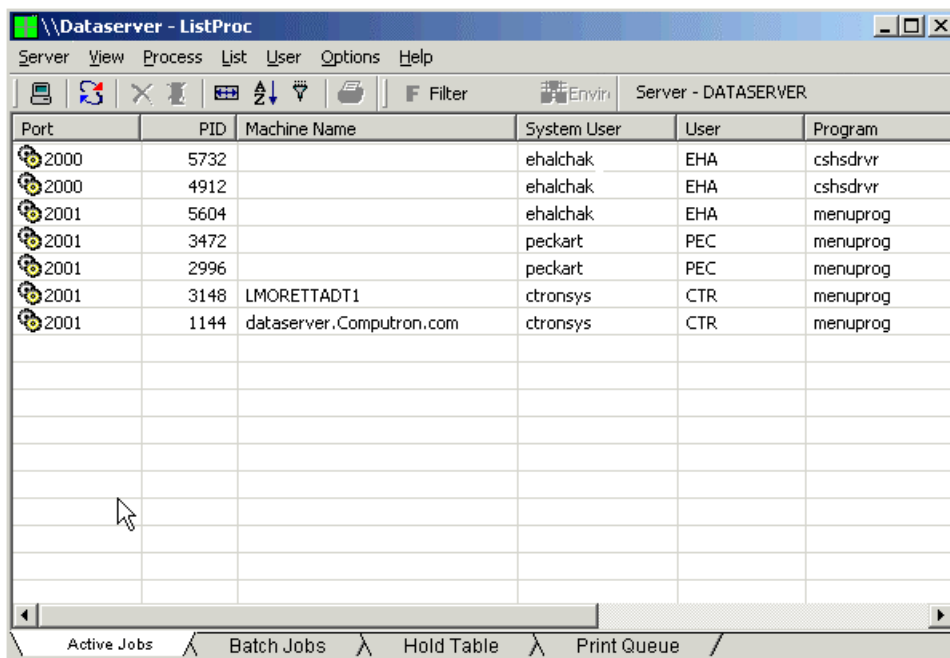
### *Introduction to Computron Process Monitor*

The Computron Process Monitor (ListProc.exe) for Windows enables you to monitor and manage active Computron processes, the batch queue, the print queue and the hold table. Using this tool, it is possible to view details for any item and modify certain attributes of certain items (i.e. set a new priority for a batch job, send a print job to a new printer or terminate a user process).

### *The Computron Process Monitor Window*



To access the Computron Process Monitor Window, click the Process Monitor icon on the Computron Server Administrator.



The screenshot shows the 'ListProc' window for 'Server - DATASERVER'. The window title is '\\Dataserver - ListProc'. The menu bar includes Server, View, Process, List, User, Options, and Help. The toolbar contains icons for refresh, stop, print, filter, and environment. The main area is a table with the following data:

Port	PID	Machine Name	System User	User	Program
2000	5732		ehalchak	EHA	cshsdrv
2000	4912		ehalchak	EHA	cshsdrv
2001	5604		ehalchak	EHA	menuprog
2001	3472		peckart	PEC	menuprog
2001	2996		peckart	PEC	menuprog
2001	3148	LMORETTADT1	ctronsys	CTR	menuprog
2001	1144	dataserver.Computron.com	ctronsys	CTR	menuprog











At the bottom of the window, there are tabs for 'Active Jobs', 'Batch Jobs', 'Hold Table', and 'Print Queue'. The 'Active Jobs' tab is currently selected.

**Computron Process Monitor Window - Displaying Active Processes**

*Note:* The window pictured here only illustrates one way to list active processes. You can determine which columns to sort by and in what order. For detailed instructions, refer to “Sort and Filtering Processes” at the end of this section.

## *Toolbar Buttons*

The following table describes each button on the toolbar.

Button	Description
	Connect to a remote server.
	Refresh the process list.
	Delete a process.
	Log a user off the system.
	Select columns to display on the Process Monitor window.
	Select the field values that you want to sort processes by.
	Select the field values that you want to filter processes by.
	Print the process listing.
	Toggles the Filter Processes tool on and off.
	Insert Environment

## *Menu Bar Commands*

Most of the menu bar commands are available as toolbar buttons or appear on shortcut menus that display when you right-click a process. To simplify things, the instructions in this section refer to toolbar buttons unless the command is only available on the menu bar.


## *Managing the Active Processes*

Upon opening the Computron Process Monitor all active processes that are currently running in either foreground or background are listed.

## Terminating Active Processes



The Computron Process Monitor helps to simplify the administrative task of terminating active processes in order to perform system maintenance.


 *Warning! Please exercise extreme caution when terminating processes; otherwise you may cause undesirable results.*

1. Select the process that you want to terminate. Click Terminate Process.
2. Click Yes to confirm your decision.
3. The process has been deleted from the system.

## Logging Users Off of a Service



The Computron Process Monitor simplifies the administrative task of logging users off of services by allowing the Server Administrator to either log off individual users or to log all users off at once.

 *Warning! Give your users ample advanced warning before logging them off of the system.*

### *Logging Individual Users Off of the Computron System*



1. Select a process containing the user's ID.
2. On the toolbar, Click Log Off User.
3. Click Yes to confirm your decision.
4. All active processes belonging to that user are terminated.

### *Logging All Users Off of the Computron System*

1. On the User menu, click Log Off and click All Users on the Submenu.
2. Select the All Users command from the submenu.
3. Click Yes to confirm your decision.
4. All processes belonging to all users have been terminated.

## Checking Active Process Details

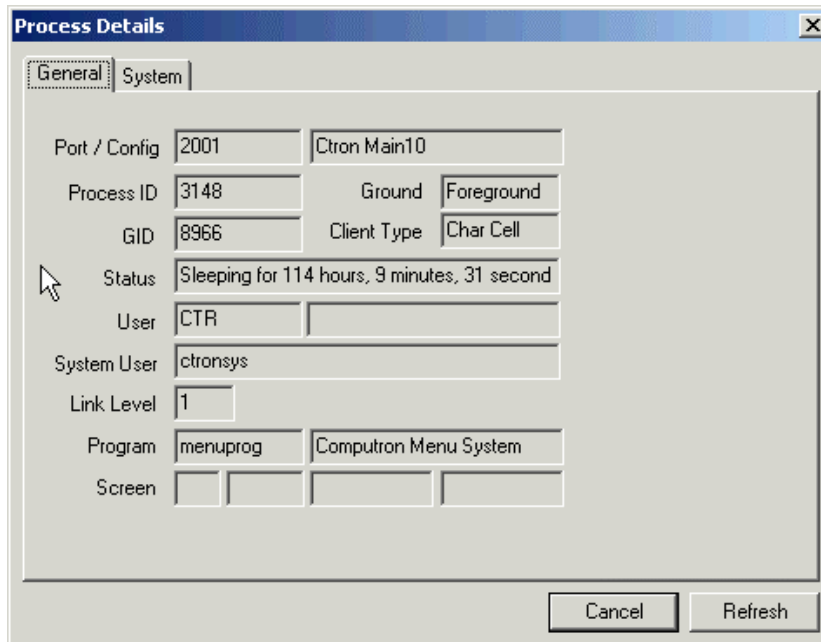
The Process Details property sheet displays with two tabbed pages of information, General and System.

1. Select the process to be viewed.
2. On the Process Menu, click Show Process Details.

☞ *Note:* Each page on the property sheet features a Refresh button that updates process detail information for that page.

### General Tab

This tab contains basic information about the process.



Process Details — General Tab

The General detail page includes the following fields:

#### Port/Config

The host port number that the process is running under.

#### Process ID

The unique ID number assigned to each process.

#### Ground

This is the background or foreground process type.

#### GID

This is the Computron master process ID for the process.

**Client Type**

This is the Computron software presentation layer.

**Status**

This is the current status of the process.

**User**

This is the Computron user ID of the user that launched the process.

**System User**

This is the Windows user ID of the user that launched the process.

**Link Level**

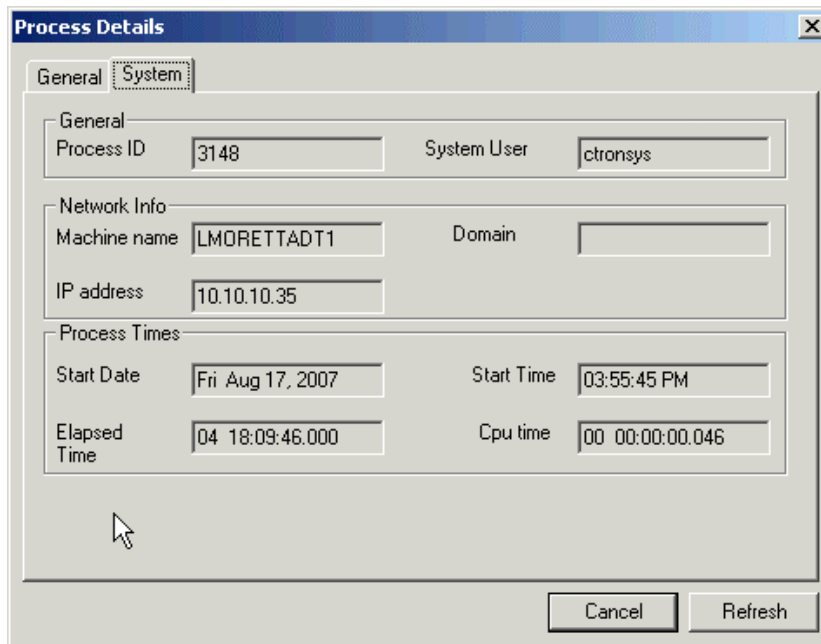
This is a number that describes the level of the process in the program stack.

**Program**

This is the application program that initiated the process.

**Screen**

This is the name of the window that is currently being displayed by the process.

*System Tab*

General	
Process ID	3148
System User	ctronsys

Network Info	
Machine name	LMORETTADT1
Domain	
IP address	10.10.10.35

Process Times	
Start Date	Fri Aug 17, 2007
Start Time	03:55:45 PM
Elapsed Time	04 18:09:46.000
Cpu time	00 00:00:00.046

Buttons: Cancel, Refresh

Process Details — System Tab

The System details page includes the following fields:

**Process ID**

This is the unique ID number assigned to each process.

---

**System User**

This is the Windows user ID of the user that launched the process.

**Machine Name**

This is the name of the client machine on which the process was accessed.

**Domain**

This is the Windows domain name (if any) or name of Ctron Server.

**IP Address**

This is the Internet Protocol address for client machine.

**Start Date**

This is the date on which the process was started.

**Start Time**

This is the time at which the process was started.

**Elapsed Time**

The amount of real time elapsed since the process was started in days, hours, minutes, seconds and thousands of seconds, respectively.

**CPU Time**

This is the amount of time that the process has actually used the system CPU.

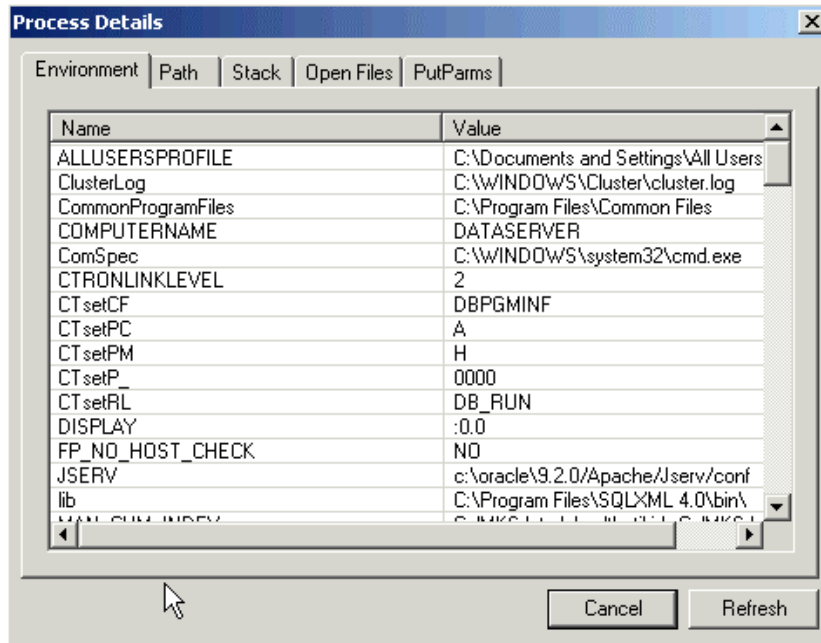
## Checking Active Process Environment

The Process Environment property sheet displays with five tabbed pages of information, Environment, Path, Stack, Open File, and PutParms.

Select the Process to be reviewed. On the Process menu, select the Show Process Environment option.

## Environment Tab

The Environment Details tab contains environment information obtained from the process currently running.



Process Details — Environment

The Environment tab details page includes the following fields:

### Name

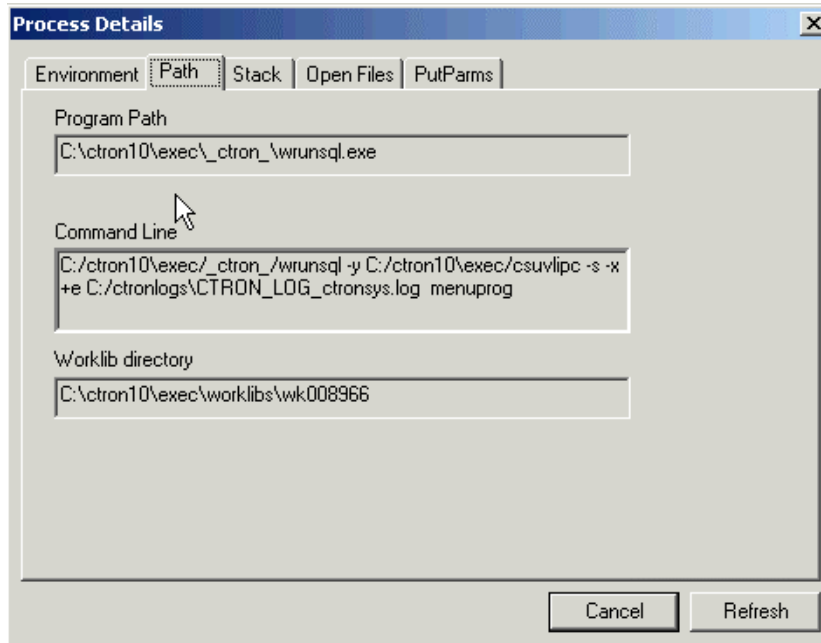
These are the names of each environment variable.

### Value

These are the values of each environment variable.

## *Path Tab*

The Path tab contains the program path, command line and worklib directory.



**Process Environment Path Tab**

The Path details tab includes the following fields:

### **Program Path**

This is the name of the program associated with the process (if known).

### **Command Line**

This is the command line which was used to start the process.

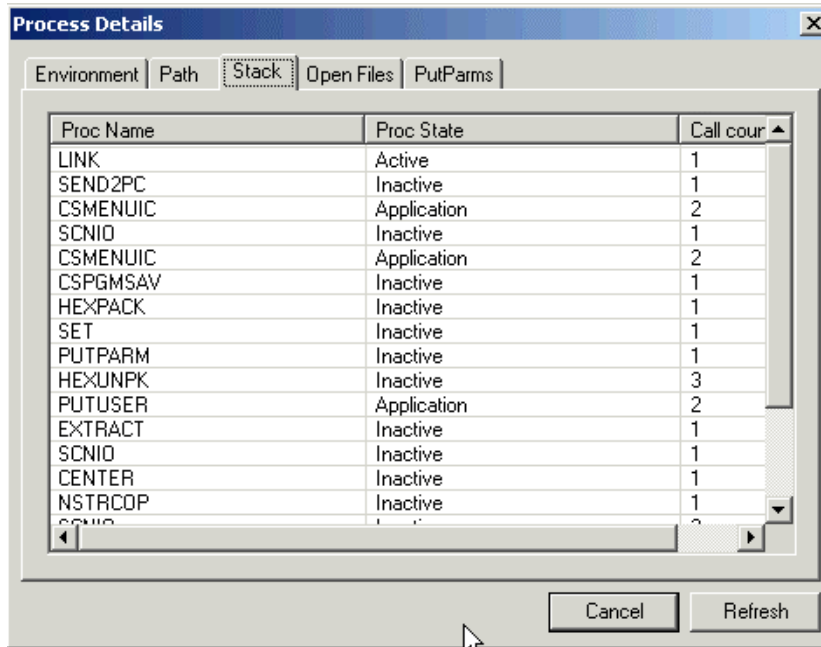
### **Worklib Directory**

This is the work library (directory) being used for the process.



## Stack Tab

The Stack tab contains processing information about active processes.



Proc Name	Proc State	Call count
LINK	Active	1
SEND2PC	Inactive	1
CSMENUIC	Application	2
SCNID	Inactive	1
CSMENUIC	Application	2
CSPGMSAV	Inactive	1
HEXPACK	Inactive	1
SET	Inactive	1
PUTPARM	Inactive	1
HEXUNPK	Inactive	3
PUTUSER	Application	2
EXTRACT	Inactive	1
SCNID	Inactive	1
CENTER	Inactive	1
NSTRCOP	Inactive	1
SCNID	Inactive	1

Process Details — Stack Tab

The Stack details tab includes the following fields:

### Proc Name

These are the names of each routine.

### Proc State

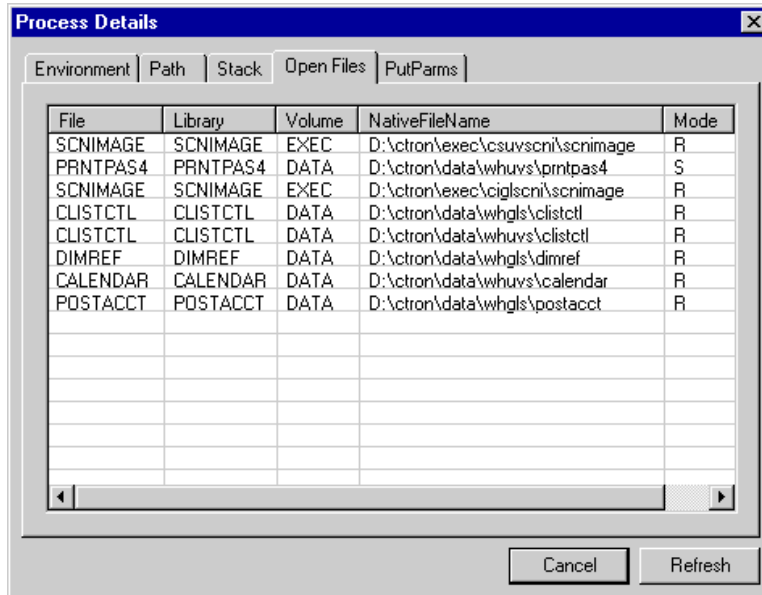
This is the current state of each routine: Active, Inactive, and Application.

### Call count

This is the number of consecutive times a routine has been called.

## Open Files Tab

The Open Files tab contains a listing of the files open on the server by the process.



File	Library	Volume	NativeFileName	Mode
SCNIMAGE	SCNIMAGE	EXEC	D:\ctron\exec\csuvscri\scnimage	R
PRNTPAS4	PRNTPAS4	DATA	D:\ctron\data\whuvs\prntpas4	S
SCNIMAGE	SCNIMAGE	EXEC	D:\ctron\exec\ciglscri\scnimage	R
CLISTCTL	CLISTCTL	DATA	D:\ctron\data\whgls\clistctl	R
CLISTCTL	CLISTCTL	DATA	D:\ctron\data\whuvs\clistctl	R
DIMREF	DIMREF	DATA	D:\ctron\data\whgls\dimref	R
CALENDAR	CALENDAR	DATA	D:\ctron\data\whuvs\calendar	R
POSTACCT	POSTACCT	DATA	D:\ctron\data\whgls\postacct	R

Environment Details — Open Files Tab

The Open Files tab includes the following fields:

### File

This is the name of the file.

### Library

This is the directory that contains the data file.

### Volume

This is a logical name (of up to six-characters) that refers to the actual location of the data directory.

### NativeFile Name

This is the actual path to the location on the server where the file resides.

### Mode

This is the mode in which the file has been accessed (e.g., read only, read/write, etc.).

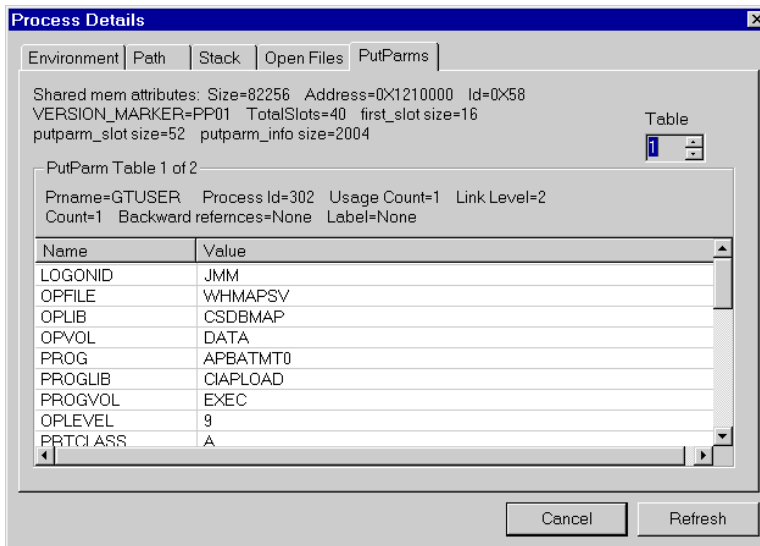
### Type

This is the type of file (e.g., indexed, sequential, etc.).

## PutParms Tab

This tab shows all of the PutParm tables that have been created for the selected process. A PutParm table is a list of information which is passed from one process to another; it includes items like user ID, program name, database map name and various other global values. PutParms are generally used to load initial data into a newly starting process, usually when one program runs another within a single application.

Viewing the information on this tab allows the administrator to see what PutParm data exists for this process and possibly diagnose problems with applications that run multiple programs.



Environment Details – PutParms Tab

The PutParms tab includes the following fields:

### Name

This is the name of the field.

### Value

This is the value of the field.

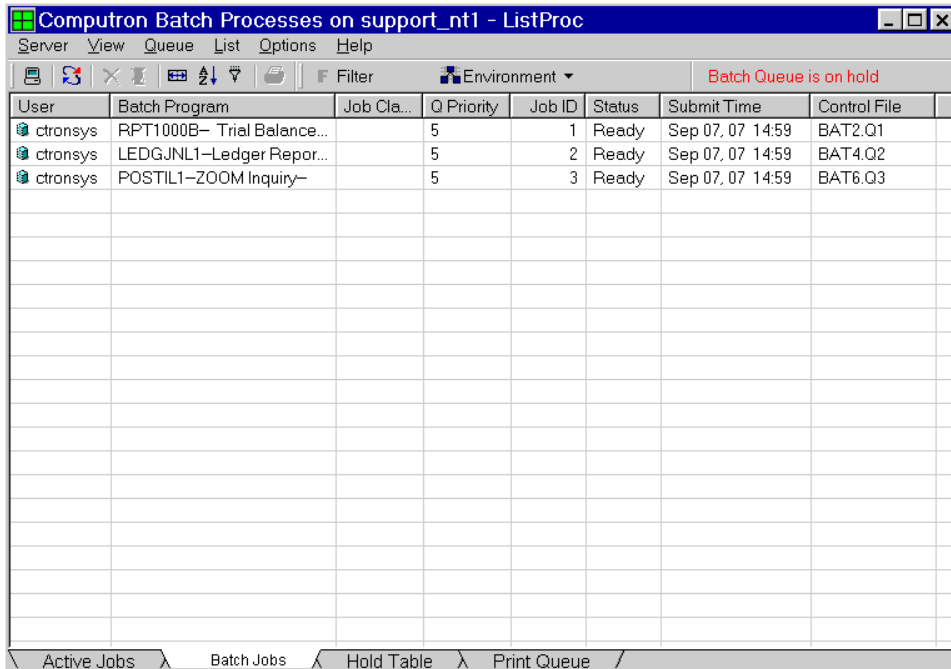
## Managing the Batch Job Processes

If the Computron Batch Queuing System is properly configured on your system, the Computron Process Monitor can also access the application processes that are currently waiting in the batch job queue or the entire job queue.

*Note:* The Computron Batch Queuing System is a background processing and queuing service designed for Computron application services running under Windows. For a high-level overview of the major concepts related to Computron's Batch Queuing system, refer to Section 5.2, "The Batch Queuing System," in this guide.

## Viewing Batch Processes

To access the batch processes in the job queue, click the Batch tab at the bottom of the Process Monitor open window.



The screenshot shows a window titled "Computron Batch Processes on support\_nt1 - ListProc". The window has a menu bar with "Server", "View", "Queue", "List", "Options", and "Help". Below the menu bar is a toolbar with various icons. A status bar at the top right indicates "Batch Queue is on hold". The main area contains a table with the following data:

User	Batch Program	Job Cla...	Q Priority	Job ID	Status	Submit Time	Control File
ctronsys	RPT1000B- Trial Balance...		5	1	Ready	Sep 07, 07 14:59	BAT2.Q1
ctronsys	LEDGJNL1-Ledger Repor...		5	2	Ready	Sep 07, 07 14:59	BAT4.Q2
ctronsys	POSTIL1-ZOOM Inquiry-		5	3	Ready	Sep 07, 07 14:59	BAT6.Q3

At the bottom of the window, there are four tabs: "Active Jobs", "Batch Jobs", "Hold Table", and "Print Queue". The "Batch Jobs" tab is currently selected.

**Computron Process Monitor Window – Batch Job Tab**

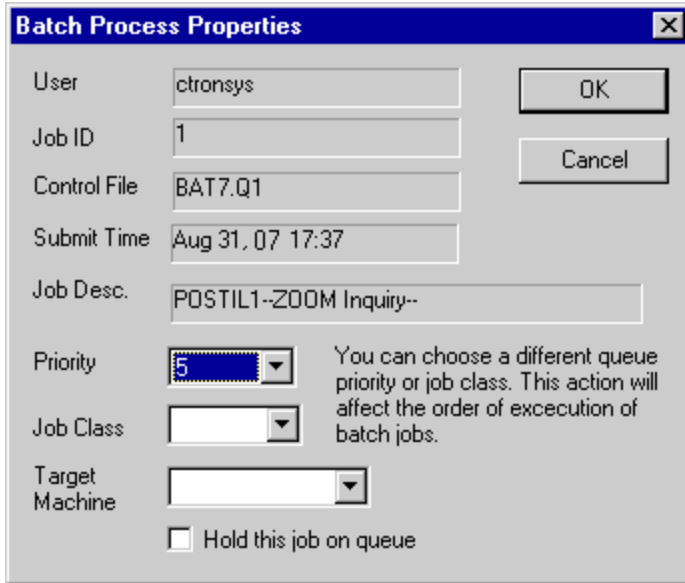
While in the Batch Job view, note that the Queue menu replaces the Process menu. Also, note that the column headings are unique to the fields that define the Computron Batch Queue System.

*Note:* The window pictured here only illustrates one way to list batch processes. You have the ability, to determine which columns to sort by and in what order. For detailed instructions, refer to “Sort and Filtering Processes” in this section.

## Accessing Batch Process Properties

This function allows the administrator to easily check the program properties of each process.

Select the process to be viewed. On the menu bar, click Queue, highlight the Item and then click Show Details on the submenu.



**Batch Process Details**

The Batch Process Properties dialog box includes the following descriptive fields:

**User**

This is the ID of the user who submitted the batch job.

**Job ID**

This is the Computron job queue ID.

**Control File**

This is a file that contains details about the batch job.

**Submit Time**

This is the time at which the batch job was submitted to the job queue.

**Job Desc.**

This is a brief description of the batch job.

The Batch Process Properties dialog box also includes the following modifiable fields, which can affect the execution of the batch jobs in the job queue. Instructions for using these options are provided in the following “Holding/Releasing Batch Jobs” section.

**Priority**

You can select another priority level by clicking the down arrow.

**Job Class**

You can select another Job Class by clicking the down arrow.

**Target Machine**

You can choose which remote server the batch job is submitted to. If not using a remote server, leave this field blank.

## Hold this job on queue

Clicking this check box holds the batch job in the job queue. Clearing the check box releases the job.

### *Changing Batch Job Priority*

You can adjust the order in which jobs are run by changing their job priority while they are waiting in the job queue. By default, all jobs are submitted with a queue priority of 5, which is the middle range of allowable queue priorities. Jobs with a higher job queue priority are run before jobs with a lower job queue priority. Jobs of equal job queue priority are processed in sequence of arrival on the queue.

1. From the Batch Process Properties dialog box, select a new priority level from the Priority list (0–9).
2. Click OK to save this setting and return to the Computron Process Monitor.

☞ *Note:* For more details about using job queue priorities effectively, see the “Job Queue Priority” heading in Section 5.4, “The Batch Queuing System.”

### *Changing the Batch Job Class*

You can change the job class of specific batch jobs, thereby assigning them to batch initiators that only accept batch jobs of a particular job class.

☞ *Note:* You can only change job classes if your batch initiators are configured to include job classes. For more details on using job classes effectively and instructions on configuring batch initiators, see Section 5.2, “The Batch Queuing System.”

1. From the Batch Process dialog box, select a new job class in the Job Class field (e.g., A, B, C, etc.).
2. Click OK.

### *Holding Specific Batch Jobs in the Queue*

You can also hold (or release) specific batch jobs in the queue regardless of the queue’s state.

1. Select the batch process to be held.
2. On the menu bar, click Queue, then select Item and click Show Details on the submenu.
3. Select the Hold this job on queue check box.
4. Click OK.

☞ *Note:* The Status for the process has changed from Ready to Held. The batch process will remain held until the Hold this job on queue check box is cleared.

## Holding/Releasing the Batch Queue

The Computron Batch Queuing System provides a single queue that receives all processes that are submitted to the background. The entire queue can be in a Held or Released state. When the queue is held, no batch processes on the queue are run until released manually; when the queue is in its released (normal) state, batch processes on the queue are run upon submission. You can also, however, hold specific batch jobs in the queue, adjust batch job release priorities and modify job classes.

### *Holding the Entire Batch Queue*

To place the entire queue in a Held state:

1. From the Queue menu, click the Hold command.
2. Click Yes when prompted to confirm whether to hold the entire job queue.

☞ *Note:* The upper right hand corner of the screen will say, “Batch Queue is on Hold.”

### *Releasing the Entire Batch Queue*

To place the entire queue in a Released state:

1. From the Queue menu, click the Release command.
2. The queued items have been released to the Batch Service for processing.

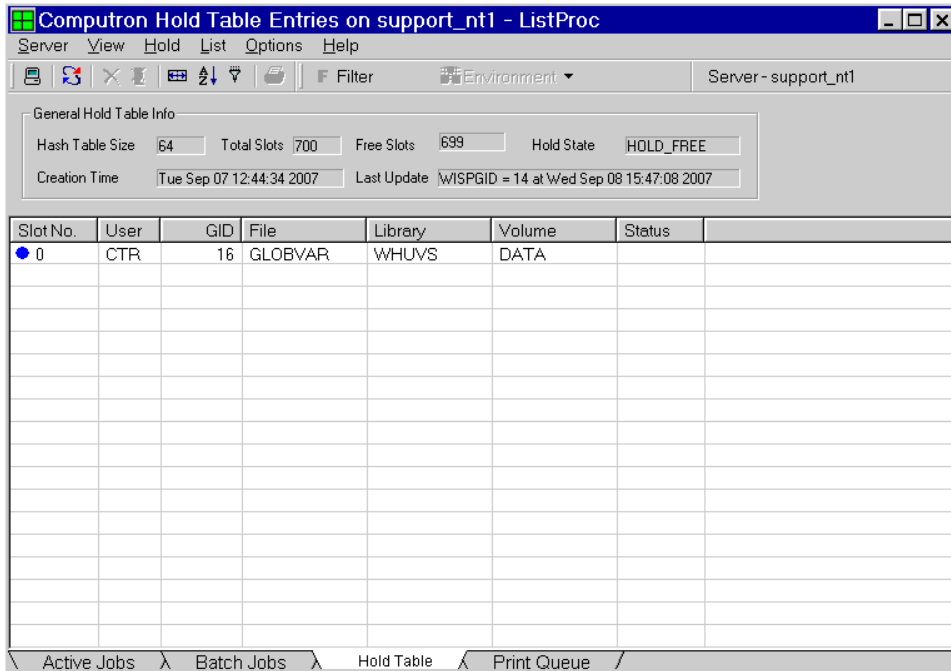
### *Deleting Batch Processes*

If necessary, batch processes can be removed from the batch queue.

1. Click the batch process that you want to delete, click Delete.
2. Click Yes when prompted to confirm deletion of the selected process.

## Managing the Hold Table Entries

Hold Table entry gives a detailed list of items currently being processed by a user. These items are held so that other users cannot access the same record. Duplicate access to a record in a file could corrupt the integrity of the file.

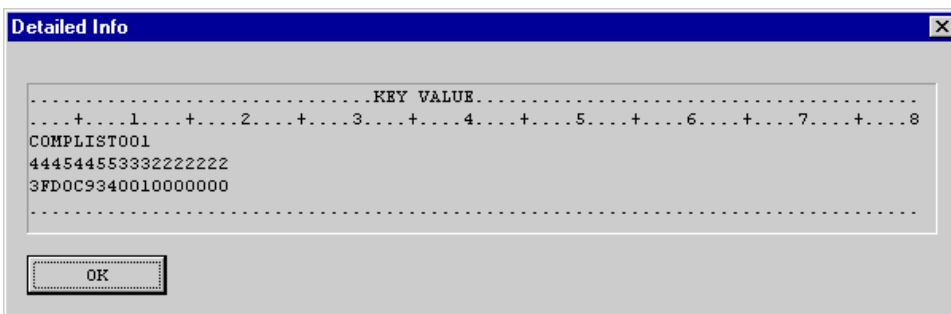


Computron Process Monitor Window – Hold Tab

### View Key Details

Key Details shows the primary key of what is being held.

1. On the Hold menu, select Entry and click Key Details on the submenu.



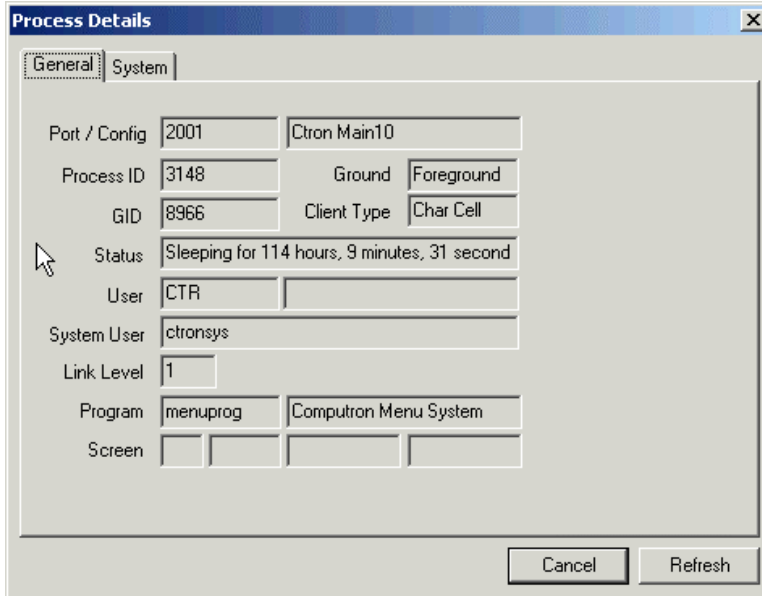
Detailed Info dialog box



## Viewing the Hold Process Details

Process Details describe the process that originally created the hold request.

1. On the Hold menu, select Entry and click Hold Process Details on the submenu.

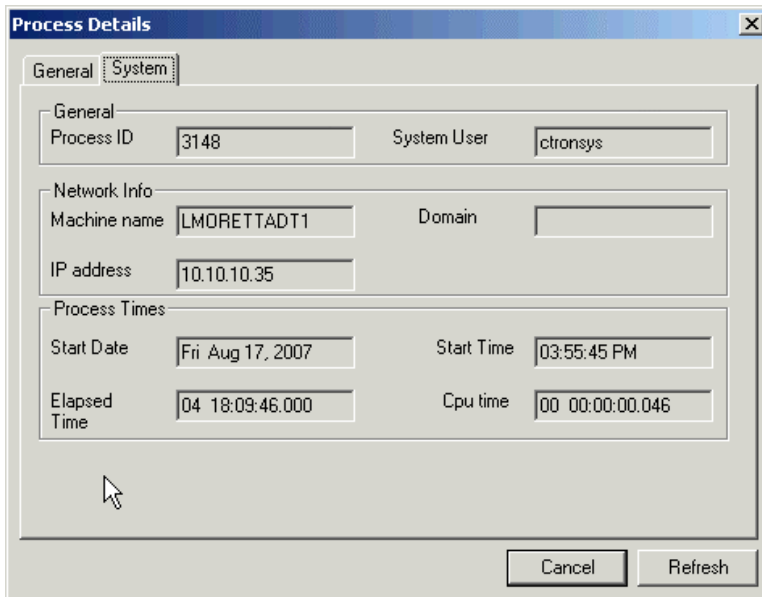


The 'Process Details' dialog box, 'General' tab, displays the following information:

Port / Config	2001	Ctron Main10	
Process ID	3148	Ground	Foreground
GID	8966	Client Type	Char Cell
Status	Sleeping for 114 hours, 9 minutes, 31 second		
User	CTR		
System User	ctronsys		
Link Level	1		
Program	menuprog	Computron Menu System	
Screen			

Buttons: Cancel, Refresh

Process Details – General Tab dialog box



The 'Process Details' dialog box, 'System' tab, displays the following information:

General			
Process ID	3148	System User	ctronsys
Network Info			
Machine name	LMORETTADT1	Domain	
IP address	10.10.10.35		
Process Times			
Start Date	Fri Aug 17, 2007	Start Time	03:55:45 PM
Elapsed Time	04 18:09:46.000	Cpu time	00 00:00:00.046

Buttons: Cancel, Refresh

Process Details – System dialog box

## Removing a Hold Entry



**Warning:** Removing a hold on a user's processes removes all holds for that user. A user should be logged off of the system before a hold is removed under their ID.

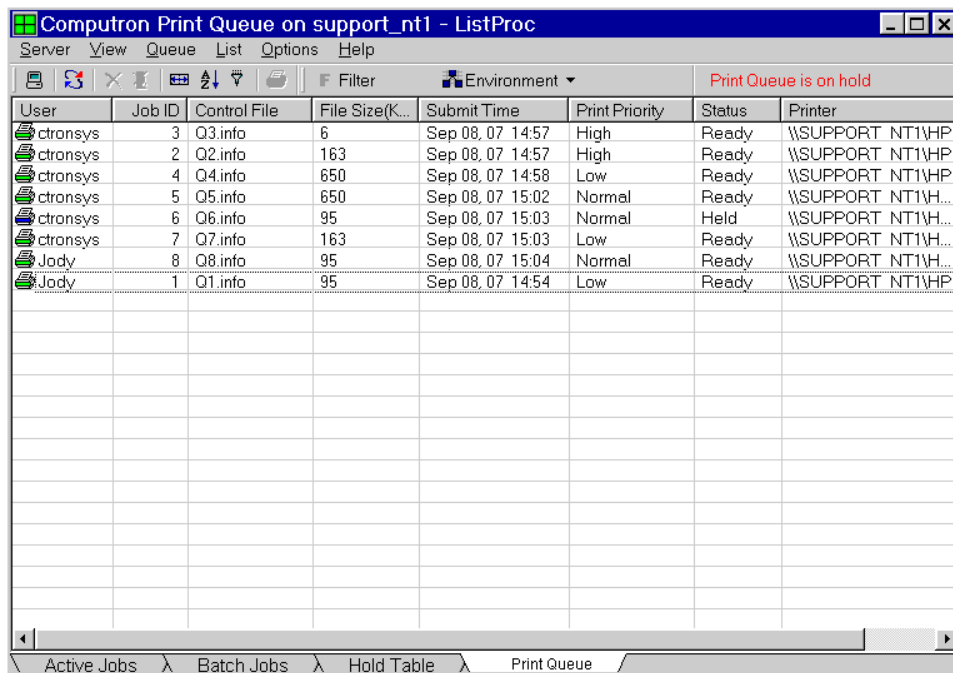
1. Click the hold entry that you want to delete, click Delete.
2. Click Yes when prompted to confirm whether to remove the hold entry.

## Managing the Print Queue Processes

If the Computron Print Queuing System is properly configured on your system, the Computron Process Monitor can also access the jobs that are currently waiting in the print queue.

### Viewing Print Jobs

To access the print queue, click the Print Queue tab at the base of the Process Monitor window.



User	Job ID	Control File	File Size(K...)	Submit Time	Print Priority	Status	Printer
ctronsys	3	Q3.info	6	Sep 08, 07 14:57	High	Ready	\\SUPPORT NT1\HP
ctronsys	2	Q2.info	163	Sep 08, 07 14:57	High	Ready	\\SUPPORT NT1\HP
ctronsys	4	Q4.info	650	Sep 08, 07 14:58	Low	Ready	\\SUPPORT NT1\HP
ctronsys	5	Q5.info	650	Sep 08, 07 15:02	Normal	Ready	\\SUPPORT NT1\H...
ctronsys	6	Q6.info	95	Sep 08, 07 15:03	Normal	Held	\\SUPPORT NT1\H...
ctronsys	7	Q7.info	163	Sep 08, 07 15:03	Low	Ready	\\SUPPORT NT1\H...
Jody	8	Q8.info	95	Sep 08, 07 15:04	Normal	Ready	\\SUPPORT NT1\H...
Jody	1	Q1.info	95	Sep 08, 07 14:54	Low	Ready	\\SUPPORT NT1\HP

**Computron Process Monitor Window – Print Tab**

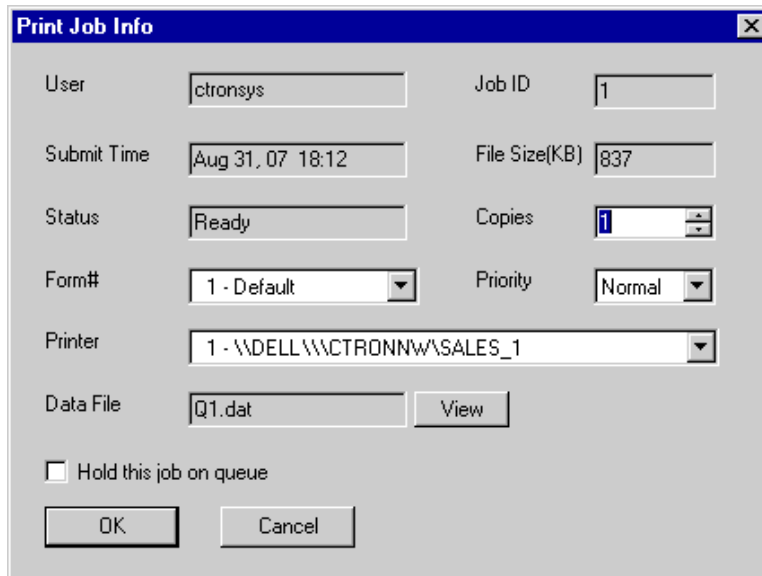
While in the Print Queue view, note that the Queue menu replaces the Process menu. Also, note that the column headings are unique to the fields that define the Computron Print Queue System.

*Note:* The window pictured here only illustrates one way to list Print Queue processes. You have the ability to determine which columns to sort by and in what order. For detailed instructions, refer to “Sort and Filtering Processes” in this section.

## *Accessing Print Job Information*

This function allows the administrator to easily check the print properties of each process.

1. Select the process to be viewed.
2. On the menu bar, click Queue, select Item and then click Show Details on the submenu.



The image shows a 'Print Job Info' dialog box with the following fields and controls:

- User:** Text box containing 'ctronsys'
- Job ID:** Text box containing '1'
- Submit Time:** Text box containing 'Aug 31, 07 18:12'
- File Size(KB):** Text box containing '837'
- Status:** Text box containing 'Ready'
- Copies:** Spin box containing '1'
- Form#:** Dropdown menu showing '1 - Default'
- Priority:** Dropdown menu showing 'Normal'
- Printer:** Dropdown menu showing '1 - \\DELL\\CTRONNW\\SALES\_1'
- Data File:** Text box containing 'Q1.dat' with a 'View' button next to it.
- Hold this job on queue
- Buttons: 'OK' and 'Cancel'

**Print Job Info dialog box**

The Print Job Info dialog box includes the following modifiable fields:

### **User**

This is the ID of the user who submitted the batch job.

### **Job ID**

This is the Computron job queue ID.

### **File Size**

This is the size of the file to be printed.

### **Submit Time**

This is the time at which the batch job was submitted to the job queue.

### **Status**

This is the description of the print job's status.

### **Data File**

This is the file that contains details about the print job.

The Print Job Info dialog box also includes the following modifiable fields, which can affect the execution of the print jobs in the print queue. Instructions for using these options are provided in the following "Holding/Releasing Batch Jobs" section.

## Copies

This is the number of copies of a document submitted to the print queue.

## Form Number

This is the form number the document is using.

## Priority

You can select another priority level by clicking the down arrow.

## Printer

This is the printer that the job has been submitted to.

## Hold this job on queue

Clicking this check box holds the print job in the queue. Clearing the check box releases the job.

### *Changing Print Job Priority*

You can adjust the order in which jobs are run by changing their priority while they are waiting in the queue. By default, all jobs are submitted with a queue priority of Normal, which is the middle value of the allowable queue priorities. Jobs with a higher queue priority are run before jobs with a lower queue priority. Jobs of equal queue priority are processed in sequence of arrival on the queue.

1. From the Print Job Info dialog box, select a new priority level from the Priority.
2. Click OK.

### *Holding Specific Print Jobs in the Queue*

Specific batch jobs can be held in the queue regardless of the queue's status.

1. Select the Print Job to be held.
2. On the menu bar, click Queue, then select Item and click Show Details on the submenu.
3. Select the Hold this job on queue check box.
4. Click OK.

☞ *Note:* The Status for the process has changed from Ready to Held. The print job will remain held until the Hold this job on queue check box is cleared.

## Holding/Releasing the Print Queue

The Computron Print Queuing System provides a single queue that receives all processes that are submitted. The entire queue can be in a Held or Released state. When the queue is held, no print jobs on the queue are run until released manually; when the queue is in its released (normal) state, print jobs on the queue are run upon submission. You can also, however, hold specific print jobs in the queue and adjust print job release priorities.

### *Holding the Entire Print Queue*

To place the entire print queue in a Held state:

1. From the Queue menu, click the Hold command.
2. Click Yes when prompted to confirm whether to hold the entire print queue.

☞ *Note:* The upper right hand corner of the window says, “Print Queue is on Hold.”

### *Releasing the Entire Print Queue*

To place the entire print queue in a Released state:

1. From the Queue menu, click the Release command.
2. The queued items have been released to the Print Queue for processing.

### **Deleting Print Jobs**



If necessary, print jobs can be removed from the print queue.

Click the print job that you want to delete, click Delete.

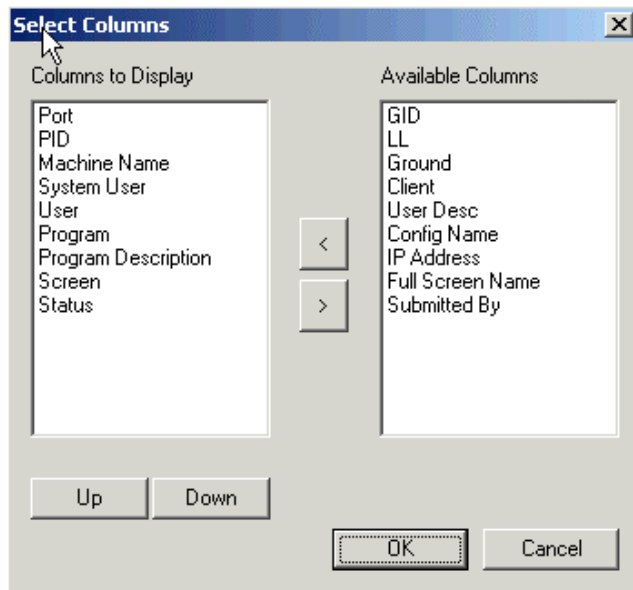
Click Yes when prompted to confirm deletion of the selected job.

## Sorting and Filtering Processes

The Computron Process Monitor allows you to choose the columns (i.e., field names) that appear on the Computron Process Monitor window. It also allows you to determine which columns to sort and/or filter processes by and in what order.

### Selecting Columns

1. From the toolbar, click Select Columns.



**Computron Process Monitor — Select Columns**

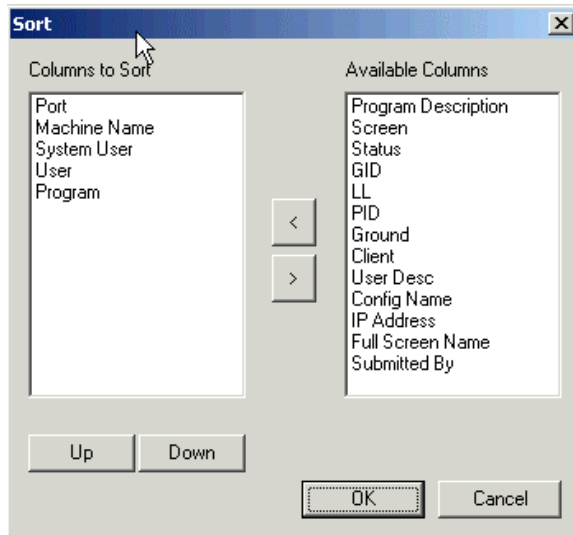
The Columns to Display box lists the columns that currently appear on the main window, while the Available Columns box lists the ones that can be added. To customize the columns:

2. Add items from Available Columns to Columns to Display.
  - ◆ Select the column name and click the left-facing arrow.
3. Rearrange column order.
  - ◆ Select the item to be moved.
  - ◆ Use the Up/Down buttons to move it to its desired position.
4. Remove a column from Columns to Display
  - ◆ Select the column name and click the right-facing arrow.
5. When all changes are complete, click OK.

*Note:* This example would display the Port column first, followed by the IP Address, Machine Name, and so on, in descending order as they appear in the Columns To Display list.

## Sorting Processes


1. From the toolbar, click Sort.



**Computron Process Monitor — Sort Columns**

The Columns to Sort box lists the columns that the listed processes are currently sorted by, while the Available Columns box lists the ones that can be added to the sort list. To customize the sorting sequence:

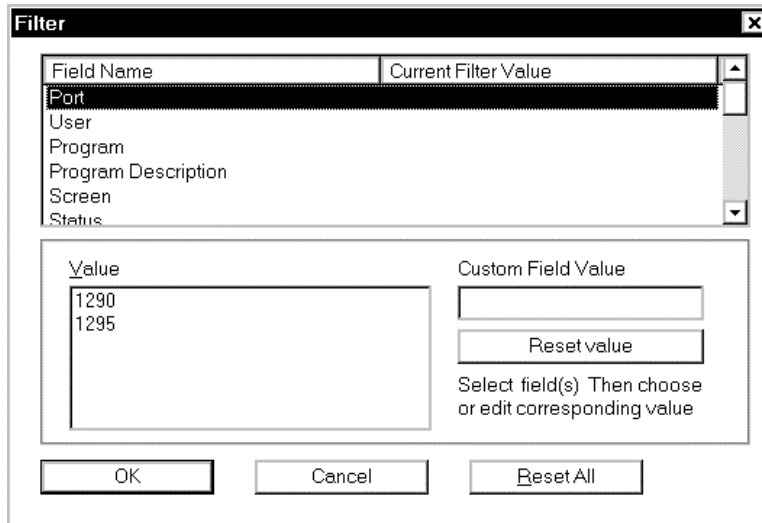
2. Add items from Available Columns to Columns to Sort.
  - ◆ Select the column name and click the left-facing arrow.
3. Rearrange column order.
  - ◆ Select the item to be moved.
  - ◆ Use the Up/Down buttons to move it to its desired position.
4. Remove a column from Columns to Sort.
  - ◆ Select the column name and click the right-facing arrow.
5. When all changes are complete, click OK.

 *Note:* The example shown here would sort processes by the Port field first, followed by the Machine Name, User ID, and Program ID fields, in descending order as they appear in the Columns To Display list.

## Filtering Processes

Using the process filter tool to condense a process list can further refine the sorting process by specific field values. The filtering tool allows fields to be filtered either by using existing filter values or by entering custom filter values. For example, if you are sorting your processes by the Port numbers, you can set the filter to view only processes for a specific Port (e.g., character cell vs. AUI).

The Process Monitor also allows you to create and maintain separate process filter for all views. Click the Filter button to create a process filter for a specific view.





Computron Process Monitor — Filter dialog box

The Field Name column entries correspond to the fields that currently appear on the Process Monitor window. When a field is selected from the Field Name column, its corresponding field value(s) displays in the Value box directly below.

To enter a field value to filter the process by:

1. Click one of the values in the Value box. The value appears in the Current Filter Value column.
2. To enter a custom value, enter it in the Custom Field Value box. For example, to only view processes for the user ID CTR, select User from the Field Name column and type CTR in the Custom Field Value box.
  - ◆ Click the Reset Value button to clear the filter value for the current field name.
  - ◆ Click the Reset All button to clear all the filter values for all field names.
3. Click OK to exit.

 *Note:* From the Process Monitor window, the toggle  button indicates whether a filter is in use and allows you to quickly switch the process filter on and off for each process view.



# **Chapter 6**

## **Database Utilities and Troubleshooting**


# 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 `ctc1prod`. 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 ctc1prod 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 `ctronsys`, with the default locations selected, has the following configuration in the `dbschema.ini` file:

```
[ rdbms=oracle database=ctcprod owner=ctronsys ]  
< 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 `ctronsys`, with the default locations selected, has the following configuration in the `dbschema.ini` file:

```
[ rdbms=sybase database=ctcprod owner=ctronsys  
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 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 `ctc1prod`. The tables being created are for the `ciap` product. The `create_tables` statement is:

```
create_tables -m ctc1prod ciapmap
```

2. The database map is `ctc1prod`. 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 -mctc1prod -v1 ciarmap aritem applhist corpcust
```

3. The database map is `ctc1prod`. 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 ctc1prod -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 `ctc1prod`. The indexes being created are for the `ciap` product. The `create_indexes` statement is:

```
create_index -m ctc1prod ciapmap
```

2. The database map is `ctc1prod`. 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 -mctc1prod -v ciarmap aritem applhist corpcust
```

3. The database map is `ctc1prod`. 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:*

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:*

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 `ctc1prod`. The stored procedure for the entire database should be dropped. Only errors are required to display. The `drop_sp` statement is:

```
drop_sp -m ctc1prod
```

## 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:*

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:*

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:*

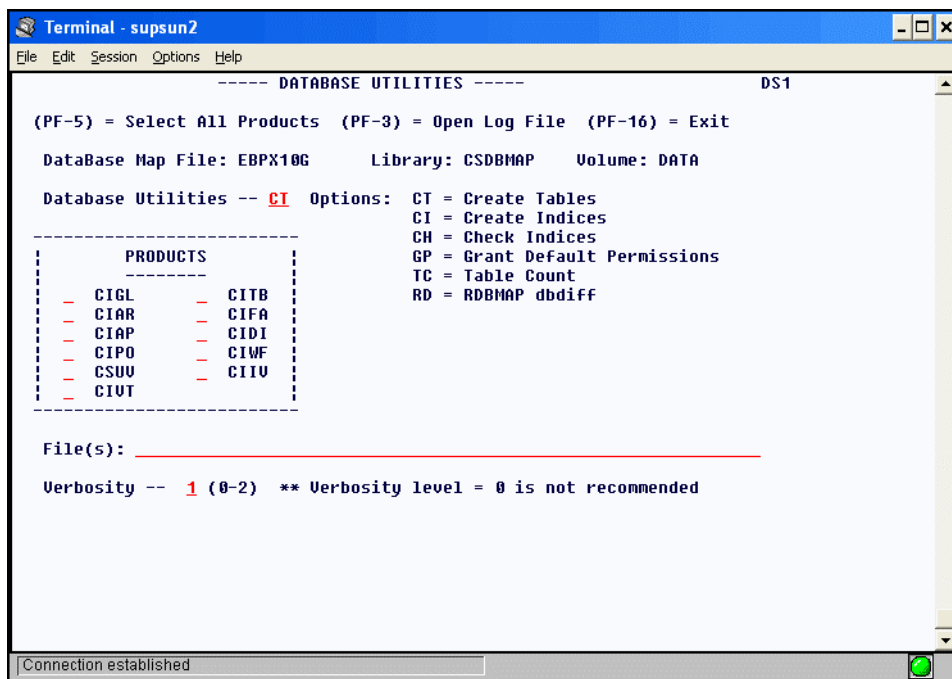
The database map is *ctclprod*. The technical consultant has a new RDBMap for the GL file, *trans* and wants to determine how much data is in the table. There are 2 tables specified in the RDBMap for this logical file. A total for both tables is needed. The *table\_count* statement is:

```
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:*

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 csputparms 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 `ctc1prod`. 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 ctc1prod -v data -l ciapdata -c 5000 ciapmap
```

2. The database map is `ctc1prod`. 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 -mctc1prod -vdata -lcigldata -c 5000 ciglmap dimref postacct
```

3. The database map is `ctc1prod`. 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 -mctc1prod -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 `cconv` program and calls several programs to generate the required `cspuparms` 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, `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 `ctc1prod`. 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 ctc1prod -v data -l ciapdata ciapmap
```

2. The database map is `ctc1prod`. 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 -mctc1prod -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.

### *Configuring a Windows Computron application server for database debugging*

Debugging produces very useful information. All options of database logging (debugging) are available in a Windows Computron application environment. Typically the Computron system administrator creates a new CSServe service. Then the administrator adds the appropriate logging entry into that service's configuration.

☞ *Note:* For information on the CSServe service see Section 5.7, “Windows Processes: A Technical Overview.”

### Creating Database Debugging CSServe Service

1. Open the Computron Server Administrator.

☞ *Note:* For information on using Computron Server Administrator, see Section 5.1, “Computron Server Administrator.”

2. Create a copy of the main CSServe for the environment where the database debug log is needed.

● *Warning!* When copying an existing service, the new service **must** use unique ports from the existing service.

☞ *Note:* For information on copying a CSServe service, see Section 5.10, “Installing a CSServe Service.”

3. Create a new configuration entry. For the “Name” enter the type of database logging (debugging) desired. For the “Value” enter “TRUE”.

☞ *Note:* For information on adding a configuration entry to a service, see Section 5.1, “Computron Server Administrator.”

4. Start the newly created service.

☞ *Note:* For information on starting a CSServe service, see Section 5.1, “Computron Server Administrator.”

5. Log into this new Service through the specified ports.

- ☞ *Note:* It is recommended that the user's log file be truncated before logging into the database debug CSServe service.
  - ☞ *Note:* For information on truncating a user's log, see Section 5.1, "Computron Server Administrator."
6. Once the process that the database logging is need for is complete, shut down the CSServe service where database logging (debugging) is enabled.

● *Warning:* It is Computron's policy that CSServe services that have database logging (debugging) enabled be kept disabled until such time when a database log is necessary. When database logging (debugging) is enabled a user's log file can fill up quickly causing undesired performance from the Computron application server.

- ☞ *Note:* For information on stopping a CSServe service, see Section 5.1, "Computron Server Administrator."

## ***Enabling Database Debugging from Within Computron Applications***

Two types of logging can be enabled from within the Computron application. "F" is for full database logging (debugging). This corresponds to the debug option CTRON\_DEBUG\_DB. "S" is for I/O statistic logging. This corresponds to the debug option CTRON\_DEBUG\_IOCOUNTER.

### **Enabling database logging (debugging) from a Computron application**

Debugging can be enabled from the Set Defaults for User Window (DS1) in character cell (PF-25 function key from a menu).

- ☞ *Note:* It is recommended that the user's log file be truncated before enabling database debug.

1. Log into a character cell session.

- ☞ *Note:* See Appendix A, for information on using character cell on a Window's based Computron application server.

● *Warning:* It is only possible to enable database debug from the application using character cell.

2. Go to the menu of the function where database logging is needed (e.g., the Accounts Payable menu).
3. Press PF-25 to enter the "set Defaults for User" window.

- ☞ *Note:* See Appendix A for information on entering PF (function) keys.

```

--- Set Defaults for User ---                               DS1
Enter Indicated Data and Press (RETURN)
--- or Press (PF-16) to Exit

Data Base Map:      EBFX10C                               in CSDBMAP                               on DATA

Default Company      Company scnio:      Batch Stream:
I/O Diagnostics     S (Off/Summary/Full)

Work/temporary files:  WORKDIR  WK006167  WORKVOL  WKVOL
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_IOCOUNT`. 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`

4. Press Enter twice to return to the application menu.

The table below provides a more detailed description of these options and information regarding when they might be used.

DEBUG OPTION	DESCRIPTION	COMMON USES
<b>CTRON_DEBUG_DB</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"> <li>◆ the application program being called,</li> <li>◆ the logical files being accessed,</li> <li>◆ the keys passed from the application and used by the runtime to access the data,</li> <li>◆ the tables being opened,</li> <li>◆ the opening mode and the method of access,</li> <li>◆ the SQL statement generated and passed to the database,</li> <li>◆ the data fetched from the database,</li> <li>◆ the configuration settings of the Computron environment and certain configurations of the RDBMS system,</li> <li>◆ other additional details.</li> </ul>	<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
<b>CTRON_DEBUG_IOCOUNT</b>	<p>This option logs statistics regarding data access. It includes:</p> <ul style="list-style-type: none"> <li>◆ the programs being run and the logical files being accessed.</li> </ul> <p>For each logical file, it includes:</p> <ul style="list-style-type: none"> <li>◆ the reads, writes, re-writes, deletes and updates made in the database related to that file.</li> <li>◆ It provides the number of rows fetched per second,</li> <li>◆ information about the hits to the runtime cache.</li> <li>◆ 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.</li> </ul>	<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>
<b>CTRON_DEBUG_SQLTRACE</b>	<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>
<b>CTRON_DEBUG_DB_CONNECT</b>	<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
<b>CTRON_DEBUG_EVALUATERULES</b>	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.
<b>CTRON_DEBUG_CSRANGE</b>	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.
<b>CTRON_DEBUG_SHOWSQL</b>	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.
<b>CTRON_DEBUG_CACHE_STATS</b>	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_IOCOUNT	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.

## 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 to 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.

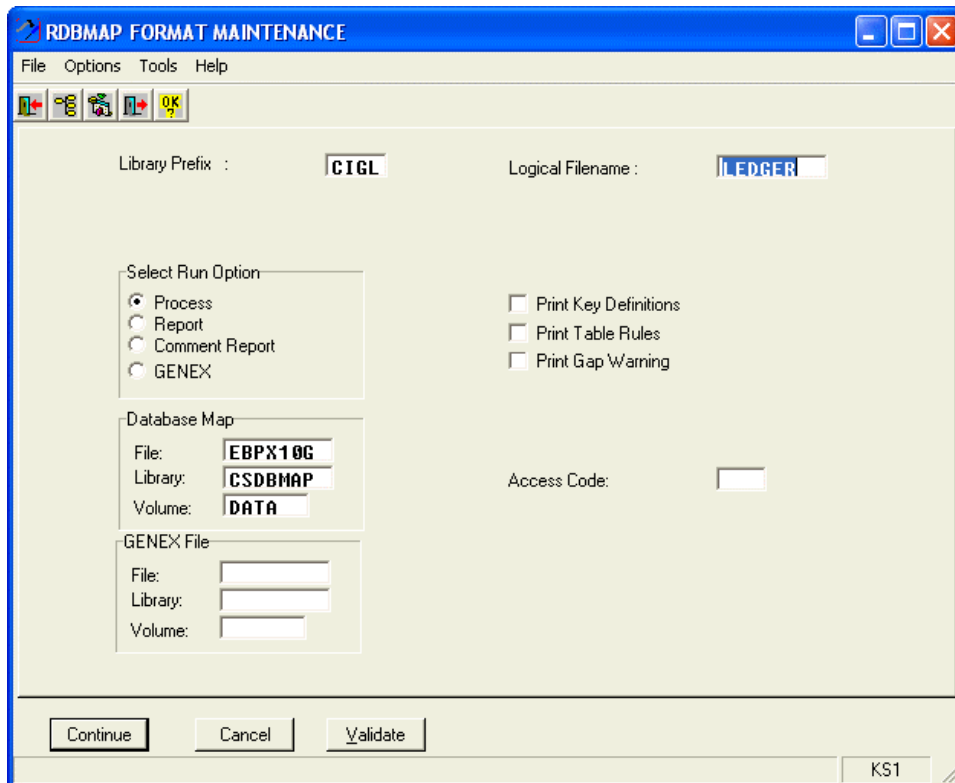
*Note:* The DDLs used by each Computron application are described in detail in the corresponding application's Technical Guide.

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:**

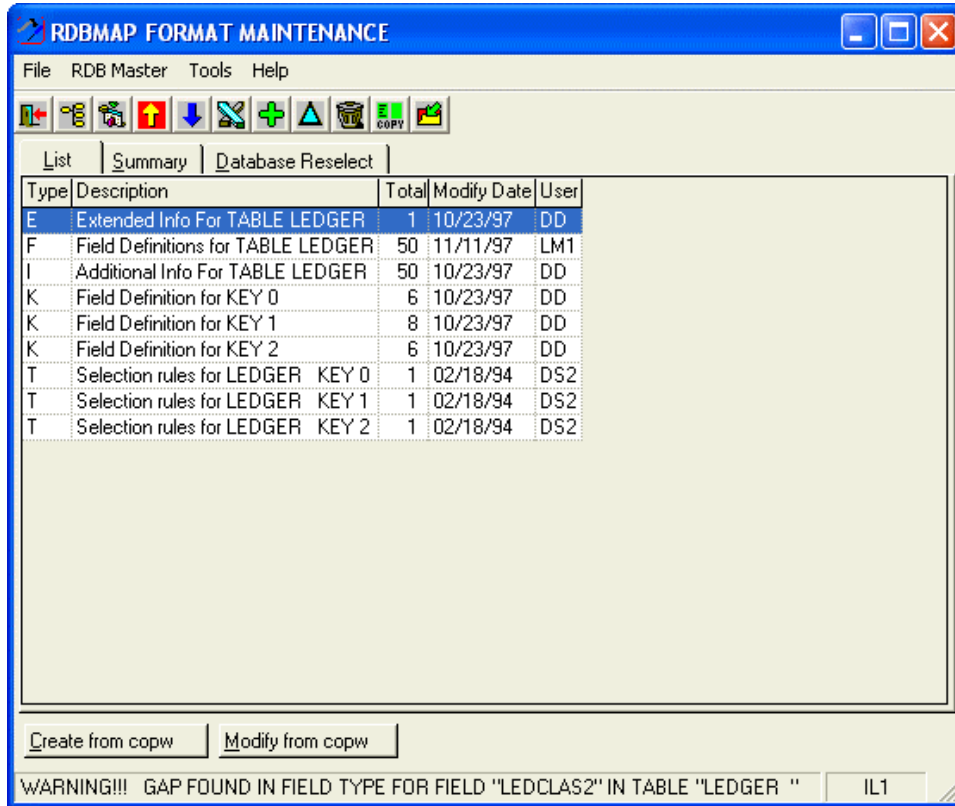
Enter the name and location of the input file to create an RDBMap. This is used when you select the GENEX option of the Select Run Option field. The file can be in either XML or flat file format. The format of the file is defined by the RDBGENEX COPW and the rdbgenex DDL or the rdbgenex XML map. The 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.

<b>Computron Options — RDBMAP Format Maintenance Inquiry/List Window</b>	
<b>Button</b>	<b>Description</b>
	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.
	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 <b>any</b> of the other functions to save changes to the RDBMap file.



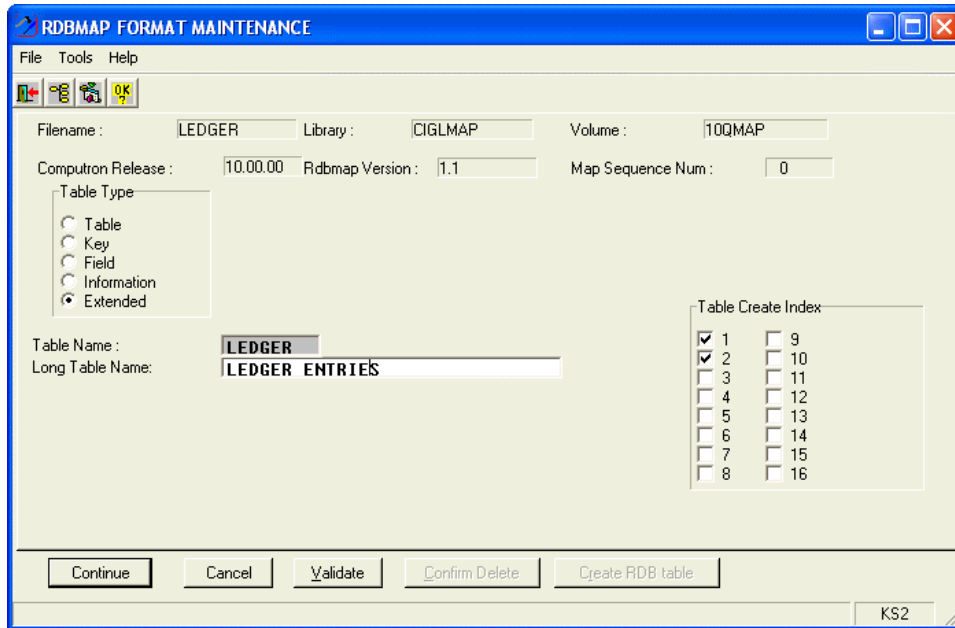
<b>Computron Options — RDBMAP Format Maintenance Inquiry/List Window</b>	
<b>Button</b>	<b>Description</b>
	<p>Creates new Key, Field, Info and Extend records for the RDBMap based on the file layout directories contained in the current Computron software environment (CIXXCOPW and CIXXCOPY [where CIXX 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

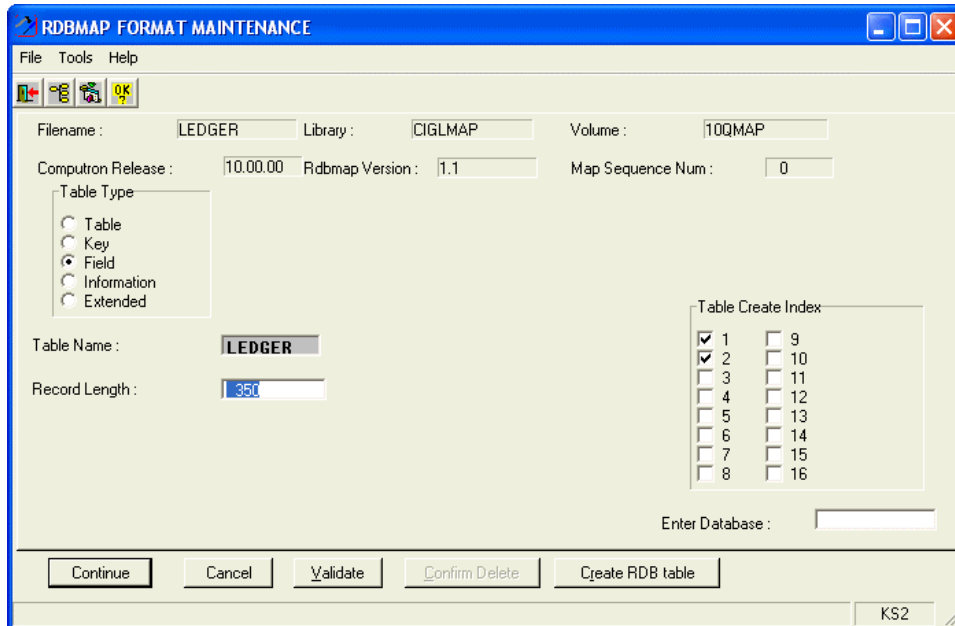
This is the eight character short table name used internally by the application and used in all other records of the RDBMap files. It provides a short-hand alias for the actual table name used in the RDBMS.

### 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 (F) 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).

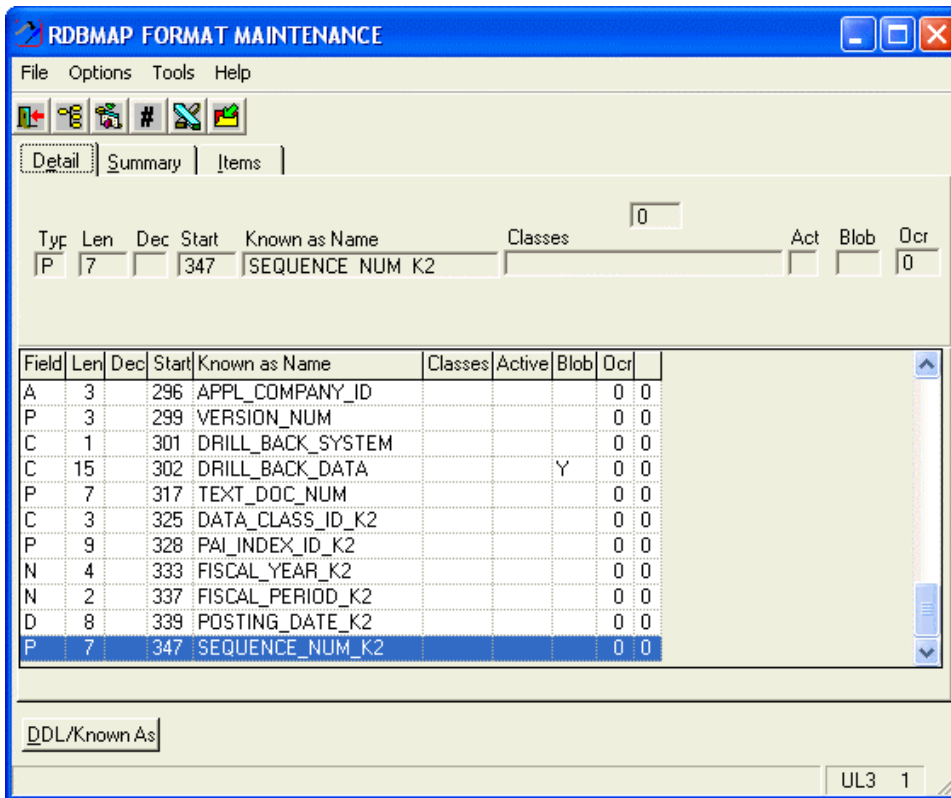
**Enter Database:**

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.



**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**

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 to XML Maps.

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**

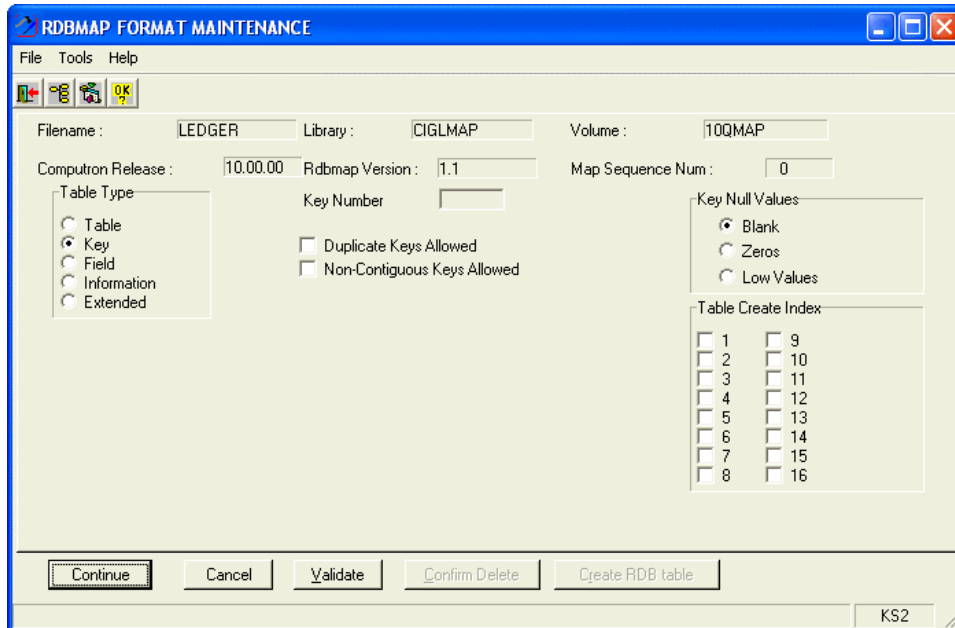
This column is not in use.

**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:

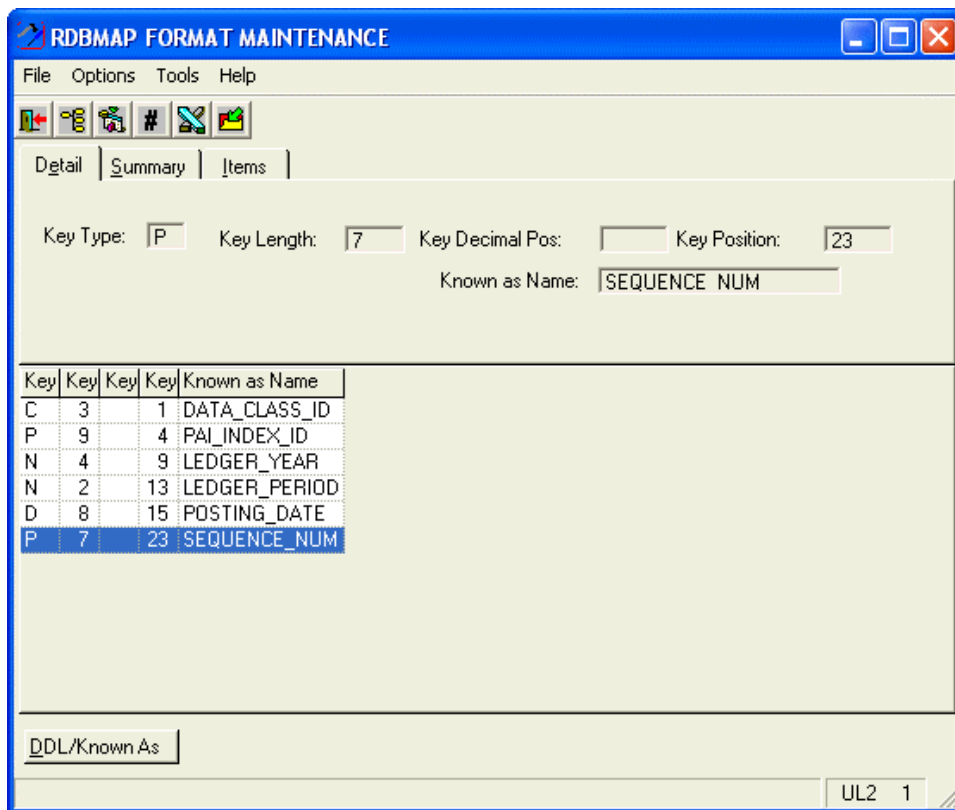
Blank – Indicates a value containing all spaces is considered null.

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.

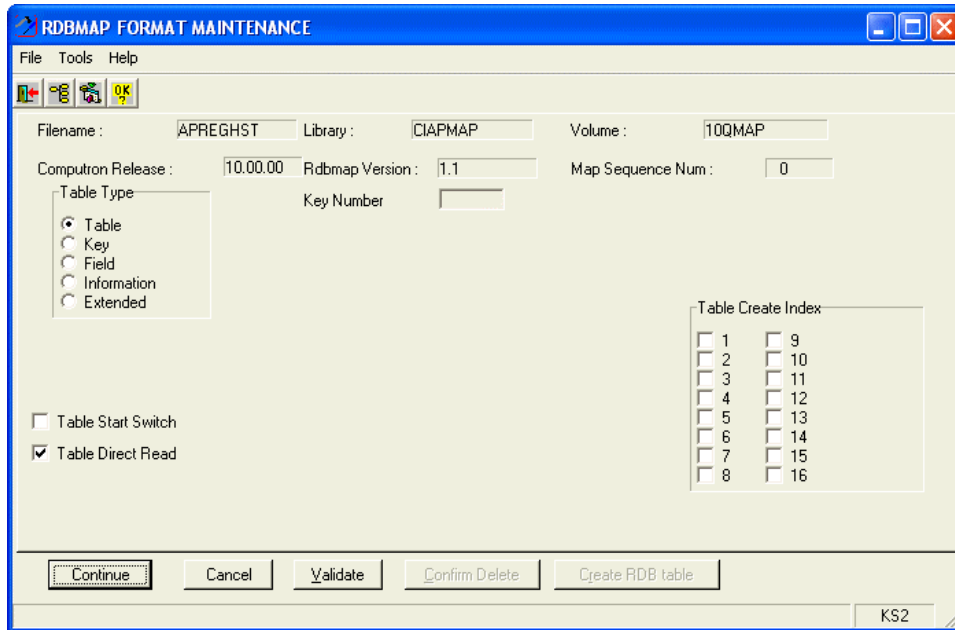
### DDL Name

This is the DDL name for the column.

## Table (T) 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.

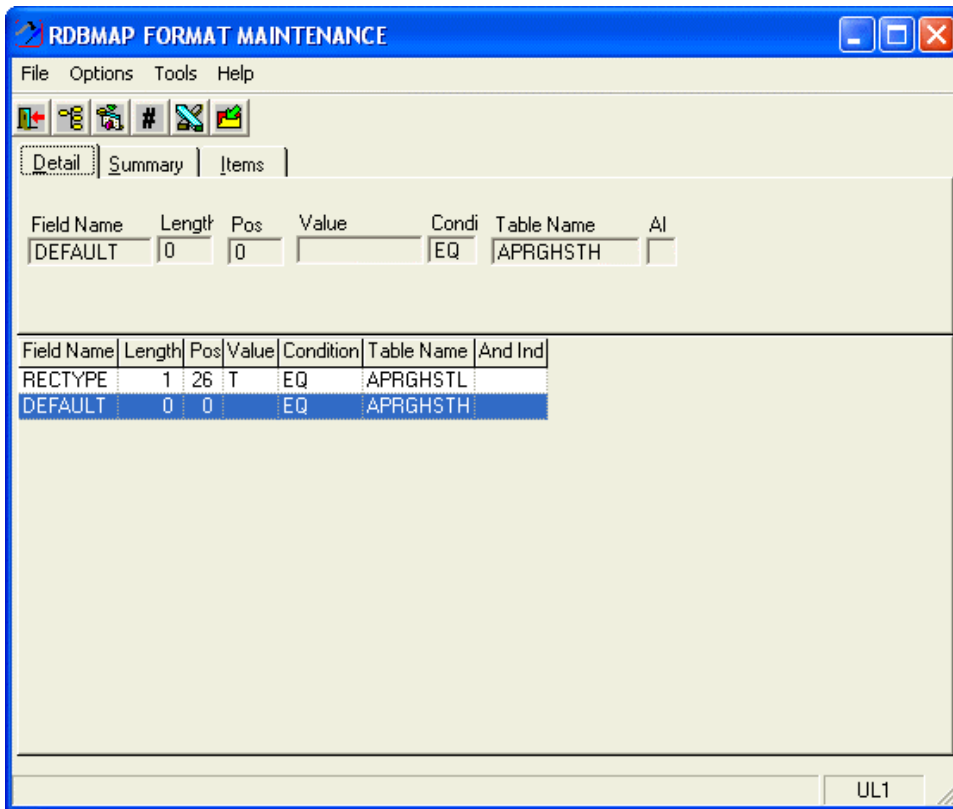
### Table Direct Read

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.

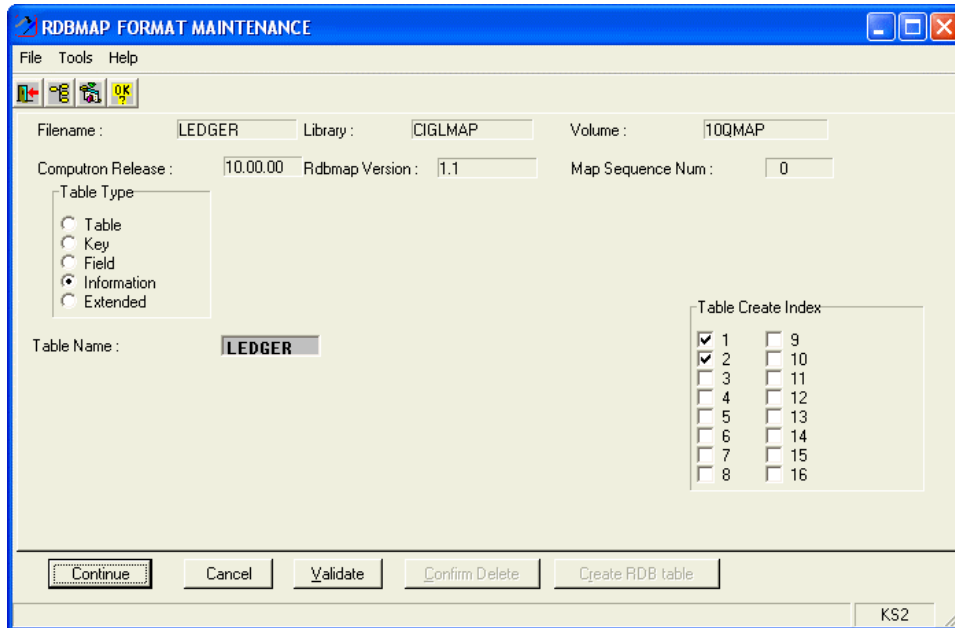
### *The Default Table Rule*

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
<b>Length</b>	0
<b>Pos</b>	0
<b>Value</b>	Blank
<b>Condition</b>	EQ
<b>Table Name</b>	Name of table to be used when no other rules is appropriate.

## Info (I) 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

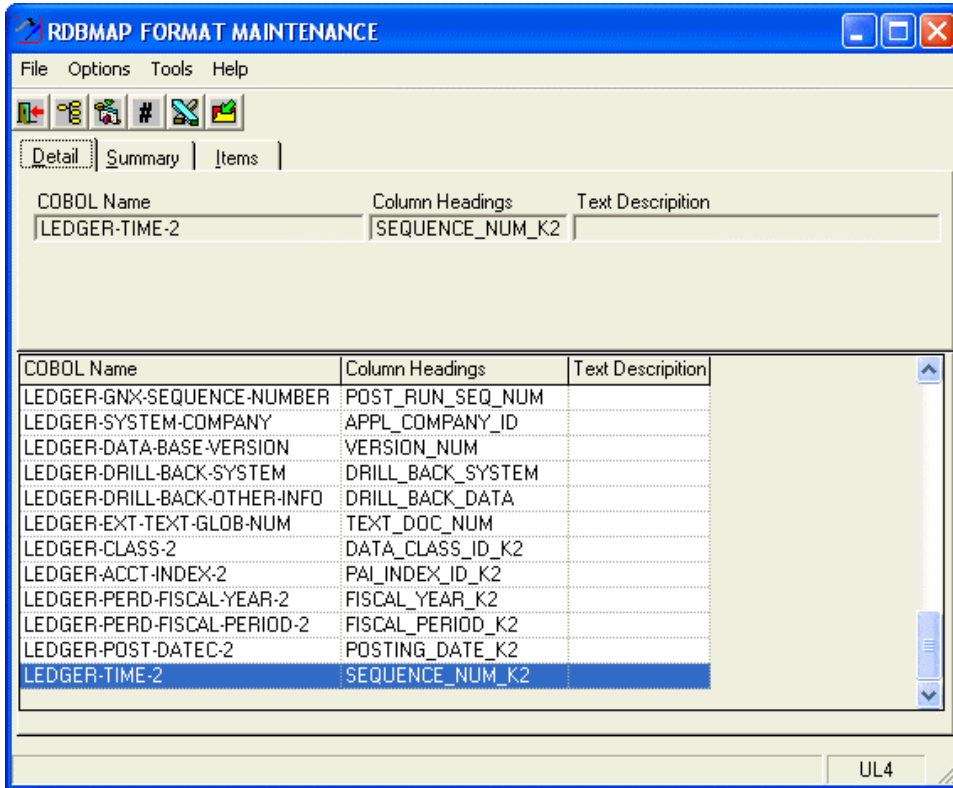
### Table Type

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.

# RDBMap Format Report Description

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	
PERDCCY1	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
TRANDTC	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	
LDGDUART	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	
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
*                                     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
PERDCCY1	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	PAI_INDEX_ID_K2	NUMBER	9
PERDCY2	FISCAL_YEAR_K2	NUMBER	4
PERDPP2	FISCAL_PERIOD_K2	NUMBER	2
PSTDTTC2	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".

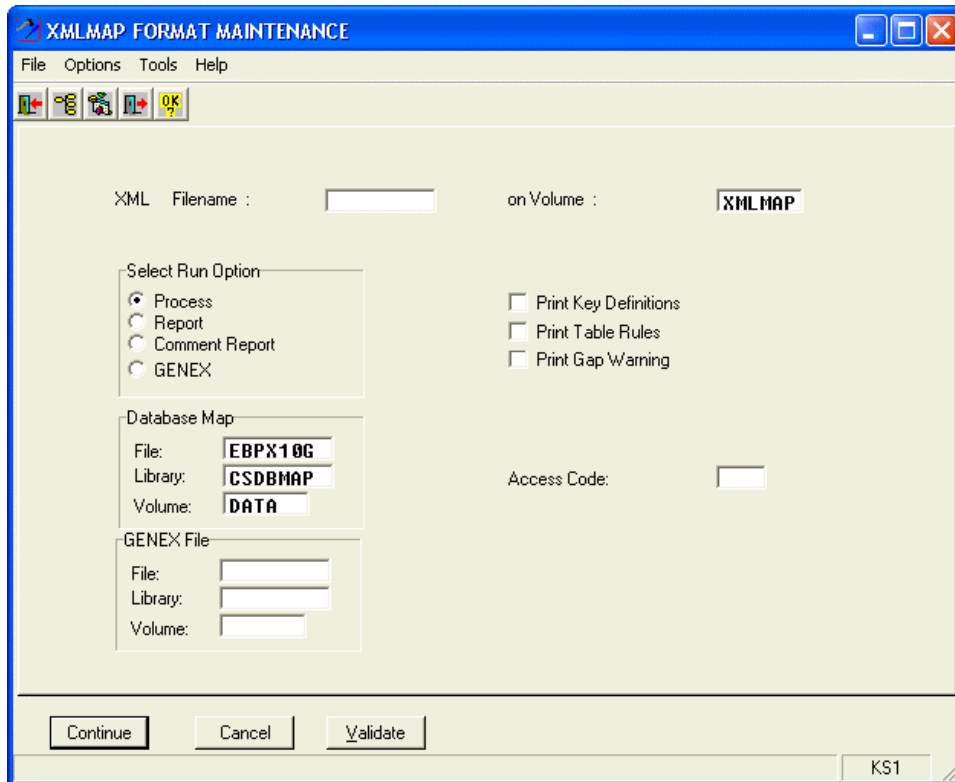
### *Introduction to XMLMap Maintenance*

Before maintaining XMLMaps, you must understand that there is a COPW and Data Description List (DDL) associated with each XMLMap. For each complete COPW/DDI 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 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.

*Default:* Process

The following fields do not apply to XML maps when the Select Run Option is set to Report.

### **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:**

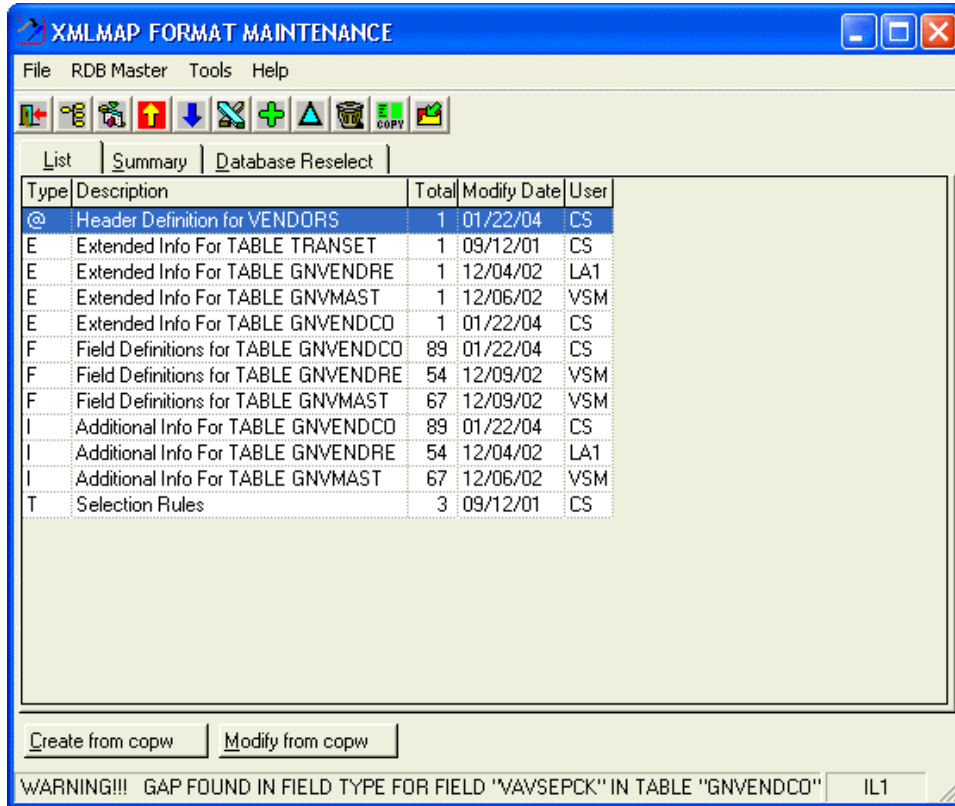
### **GENEX File: / Library: / Volume:**

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:

<b>Record Type</b>	<b>Description</b>
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 showing a warning: "WARNING!!! GAP FOUND IN FIELD TYPE FOR FIELD 'VAVSEPC' IN TABLE 'GNVENDCO'" and the window title "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.

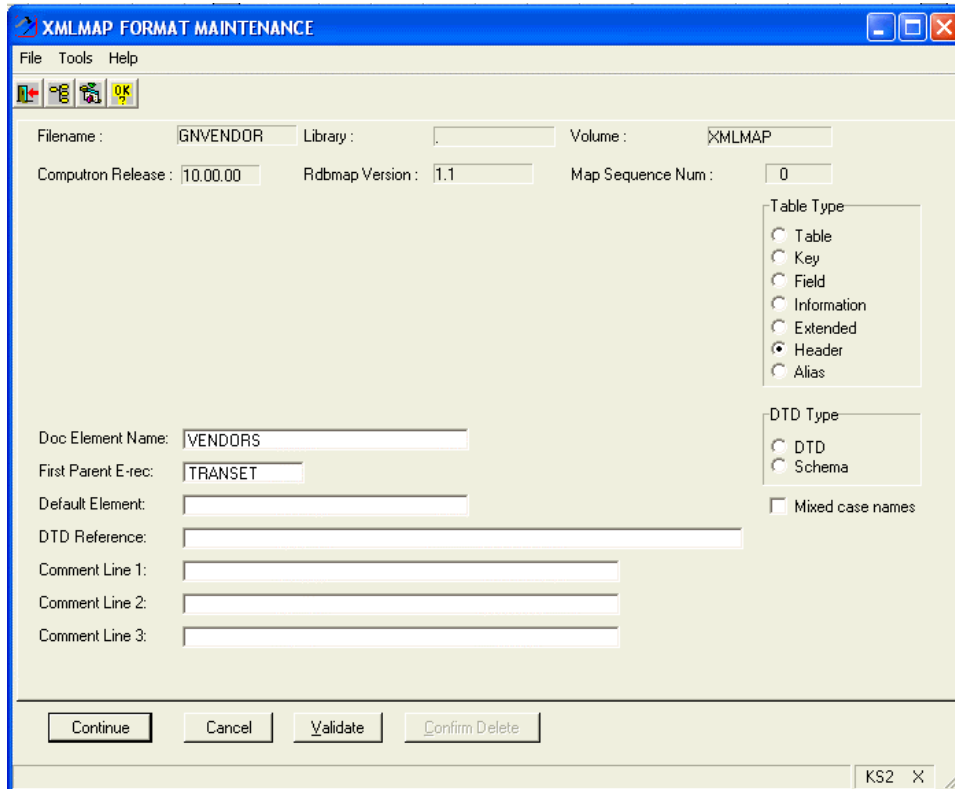
Options on the XMLMAP Format Maintenance Inquiry/List Window	
Button	Description
	Exit – This option exits the IL1 window of XMLMAP Format Maintenance.
	Previous – Returns to the previous window without exiting the XMLMAP function.
	More Data – Loads additional records in the UL1 window.
	Create new 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"> <li>◆ 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.</li> <li>◆ 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.</li> </ul>
	<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
- 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)
- DTD Type:
  - DTD
  - Schema
- Mixed case names

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.

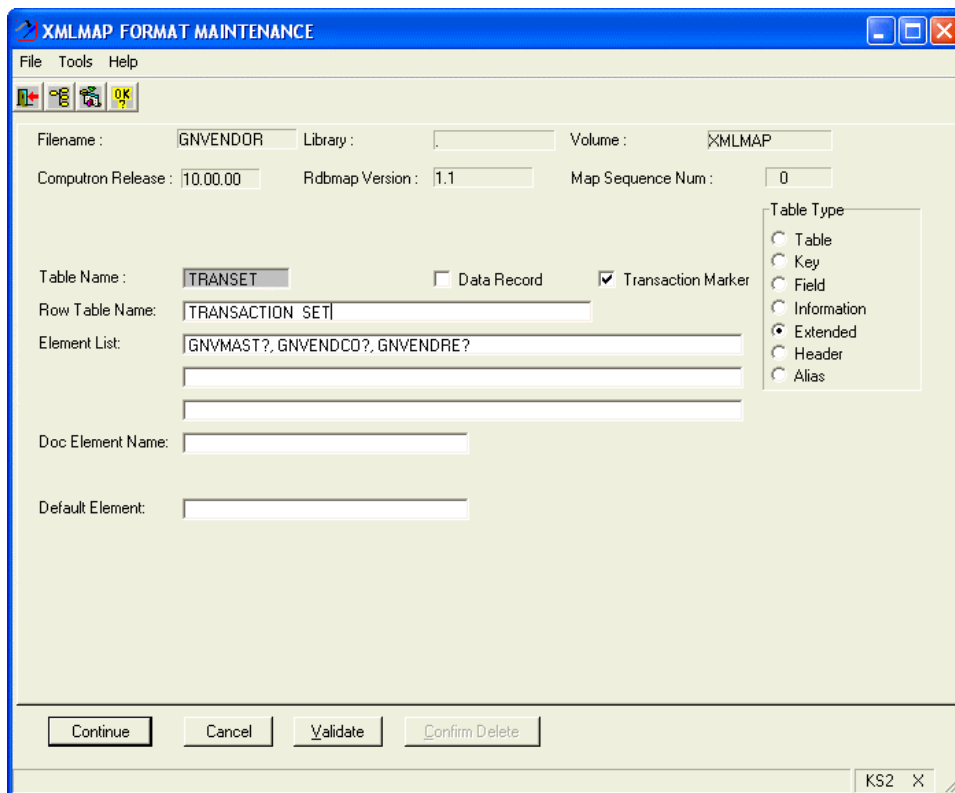
## Extended (E) Records

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 with the following fields and options:

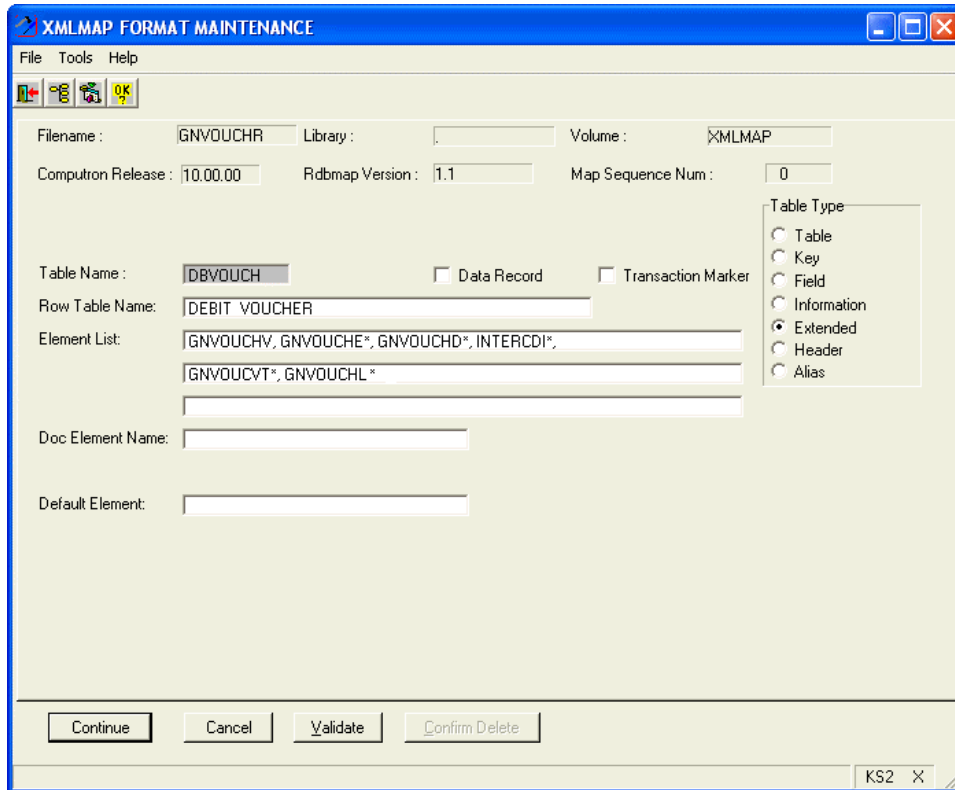
- Filename: GNVENDOR
- Library: .
- Volume: XMLMAP
- Computron Release: 10.00.00
- Rdbmap Version: 1.1
- Map Sequence Num: 0
- Table Name: TRANSET
- Data Record
- Transaction Marker
- Row Table Name: TRANSACTION SET
- Element List: GNVMAST?, GNVENDCO?, GNVENDRE?
- Doc Element Name: (empty)
- Default Element: (empty)
- Table Type:  Table,  Key,  Field,  Information,  Extended,  Header,  Alias

Buttons at the bottom: Continue, Cancel, Validate, Confirm Delete.

**XMLMAP Format Maintenance Window (KS2 X) – TRANSET Extended Record**

- ◆ The second type of Extended record contains the list of formats that make up 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 GNVOUCHR 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

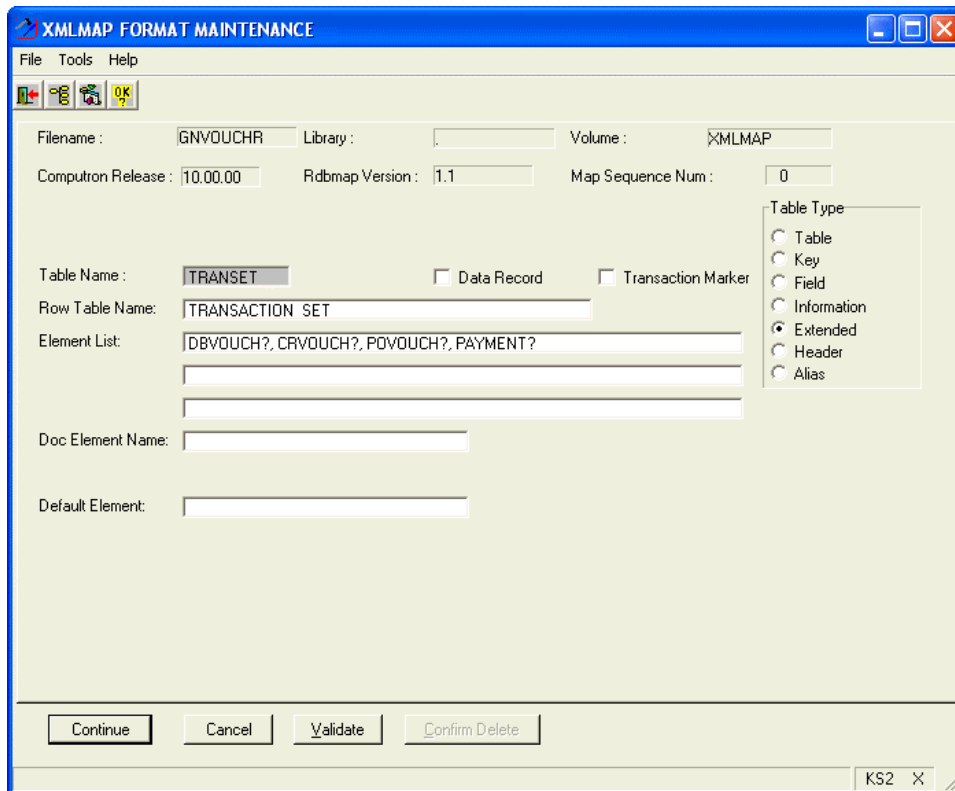


multiple records each containing some of the data for a complete GENEX input. Sometimes, the difference between one format and another is that they are different records for different files, e.g., the GENEX file for vendors, assets, customer, etc. Sometimes the difference between one format and another 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 GNVOUCHR, 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 GNVOUCHV, GNVOUCH\*, GNVOUCHD\*, INTERCDI\*, GNVOUNCVT\*, GNVOUCHL\*. All except the first one is optional.

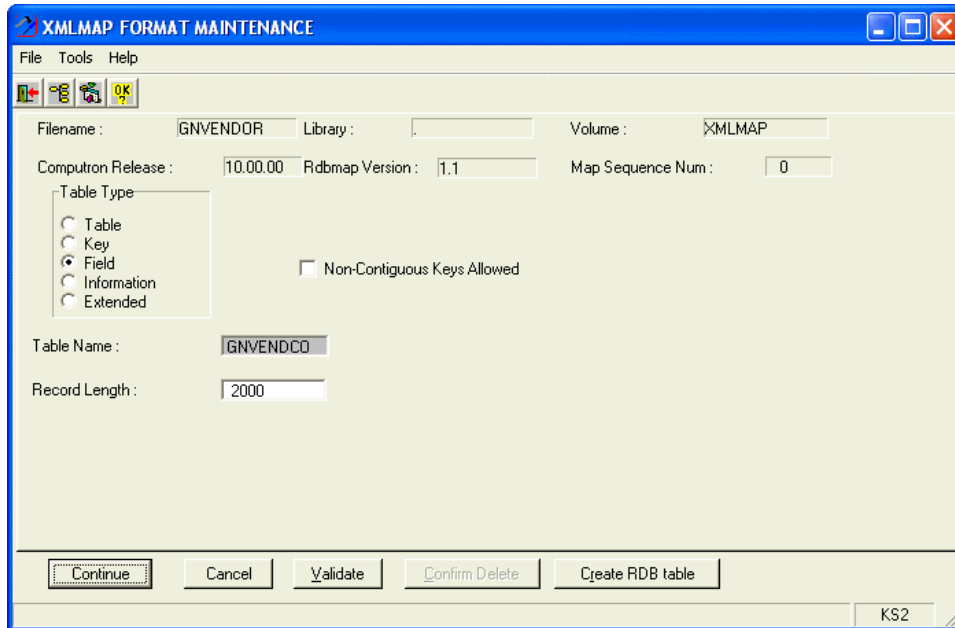


The screenshot shows the 'XMLMAP FORMAT MAINTENANCE' window. The 'Table Name' field is set to 'TRANSET'. The 'Row Table Name' is 'TRANSACTION SET'. The 'Element List' field contains 'DBVOUCH?, CRVOUCH?, POVOUCH?, PAYMENT?'. The 'Table Type' dropdown is set to 'Extended'. The 'Data Record' and 'Transaction Marker' checkboxes are unchecked. The 'Doc Element Name' and 'Default Element' fields are empty. The window has a menu bar with 'File', 'Tools', and 'Help'. The status bar at the bottom right shows 'KS2 X'.

XMLMAP Format Maintenance Window (KS2) – TRANSET Extended Record

## Field (F) 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

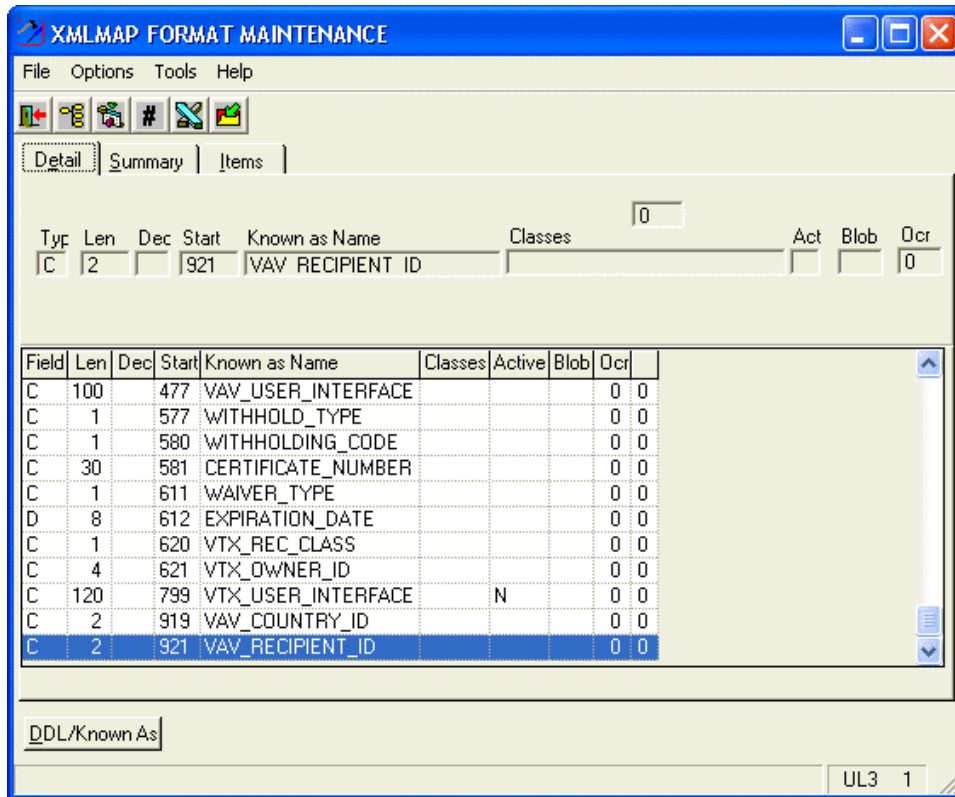
This is the eight character name of the GENEX format for this XMLMap.

### 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

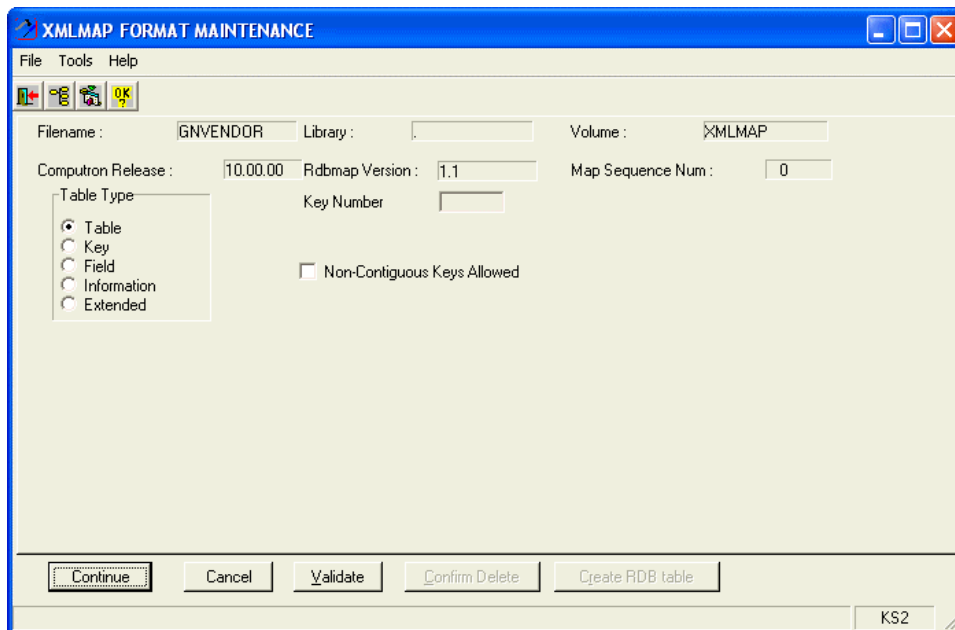
This is not used for GENEX files.

### 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 (T) Rules

Table rules are used by the system to tie the record type to the record format.

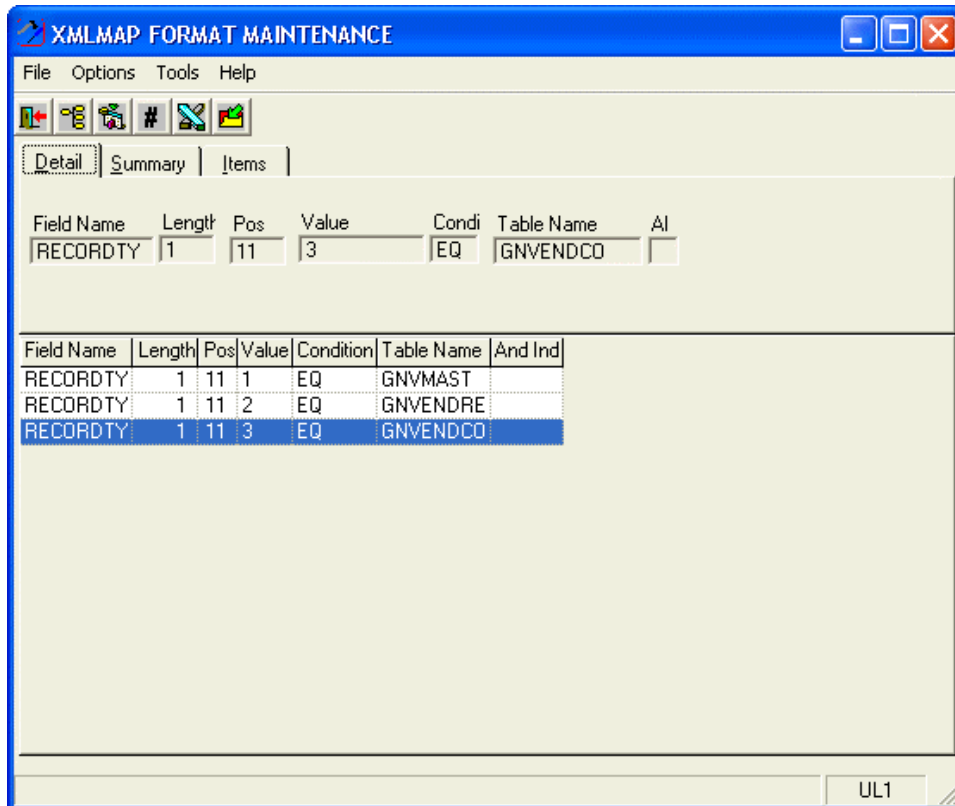


XMLMAP Format Maintenance Window (KS2) – Table Records

### 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.

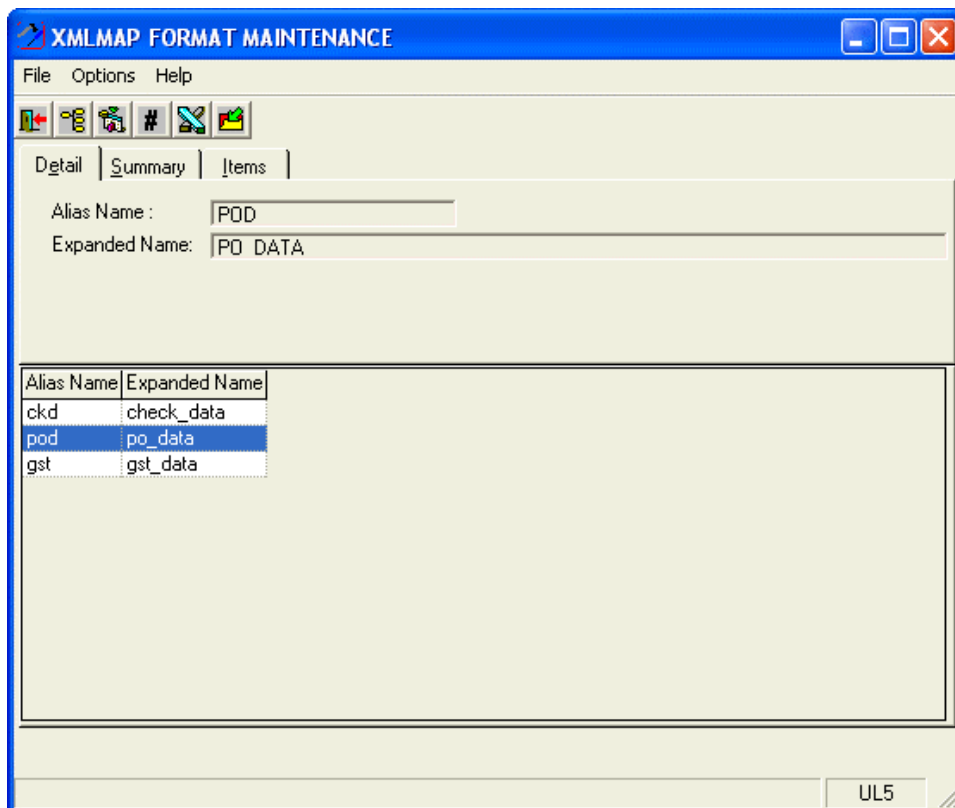
## Info (I) Records

Although Information records are displayed in the XMLMap, GENEX functions do not use this information.

## Alias (A) 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

# XMLMap Format Report Description

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_acct_ind?, chk_cr_bal_b4_pmt?,
  check_corp_bal_ind?, num_invs_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?, rlse_security_lvl1?, 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_invc_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 rise_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 of UEXMAP Format Maintenance*

The Computron UEXMAP files serve as external schema that defines the required structure of runtime UserExit 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 Collecting Information using the Computron Server Administrator

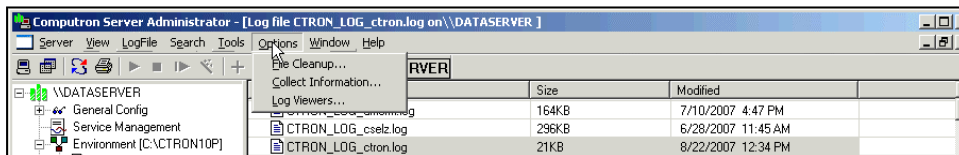
## *Introduction to Collecting Information*

Collect Information is a feature within the Computron Server Administrator. It is used to collect information about a Computron application server. The types of information collected can range from Server system settings to Computron application program versions. Once this information is collected it can be used to analyze the Computron server environment. It can also be forwarded to Computron support.

*Note:* For information on the Computron Server Administrator, see Section 5.1, “Computron Server Administrator.”

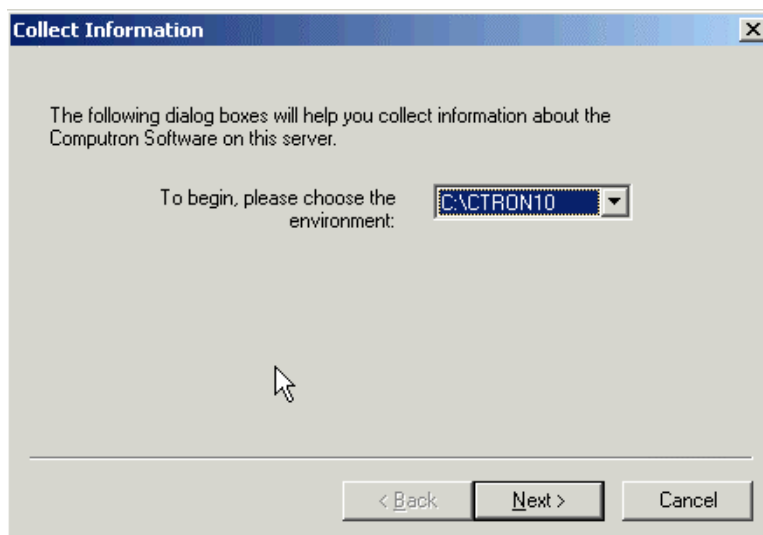
## *Using “Collect Information”*

1. From the Options menu in the Computron Server administrator, select the Collect Information option.



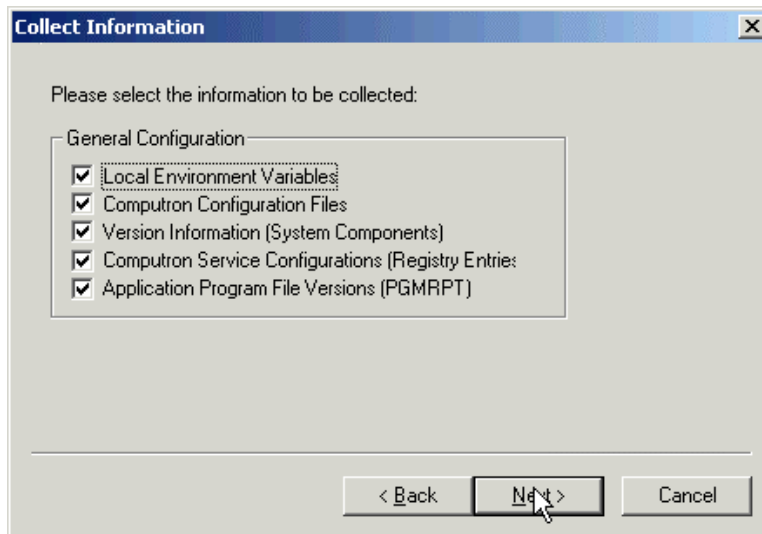
**Collect Information menu selection**

2. On the Collect Information Wizard first window, use the pull down menu to select the environment from which the information is desired.



**Environment selection window**

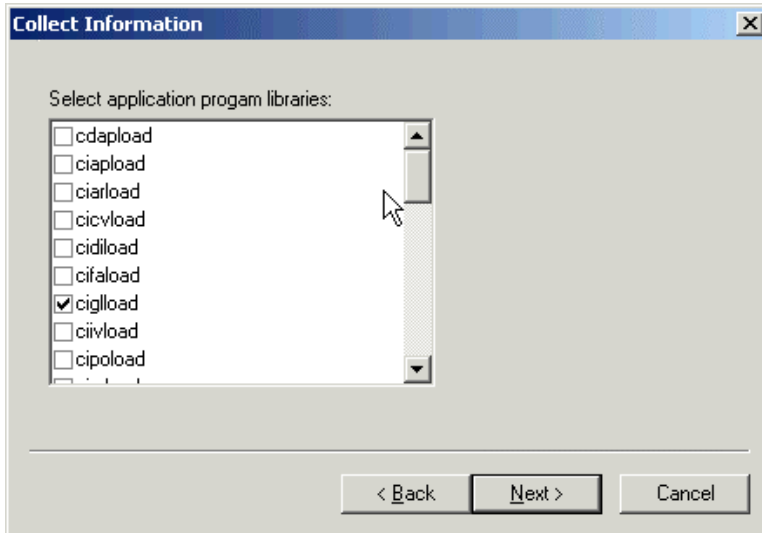
3. On the Collect Information Wizard second window, place a check mark next to the item (or items) where information is needed.



**Please select the information to be collected**

- a) Local Environment Variables – This item gathers information about the server’s settings.
    - ☞ *Note:* This selection is the similar to running “set” from a DOS prompt.
    - ☞ *Note:* This selection can only be made when running Collect Information from the Computron application server. When Collect Information is run from a remote machine, this item is grayed out.
  - b) Computron Configuration Files – This reads and displays the entries in various Computron application configuration files. Including: LGMAP, .config, ACUCONFIG, etc.
  - c) Version Information (System Components) – This selection collects versions of various Computron “technology” components (i.e. runtime, configman.exe, listproc.exe, etc).
  - d) Computron Service Configurations (Registry Entries) – This item collects values of all Computron server related registry entries.
  - e) Application Program File Versions (PGMRPT) – This entry gathers the versions of all programs in the selected Computron application directories.
    - ☞ *Note:* For more information on running a Program Report list, see the section titled “Program Report List”.
4. Click Next
  5. If “Application Program File Versions” was not selected skip to step 8
  6. Select the “Load” directories that versions are necessary.

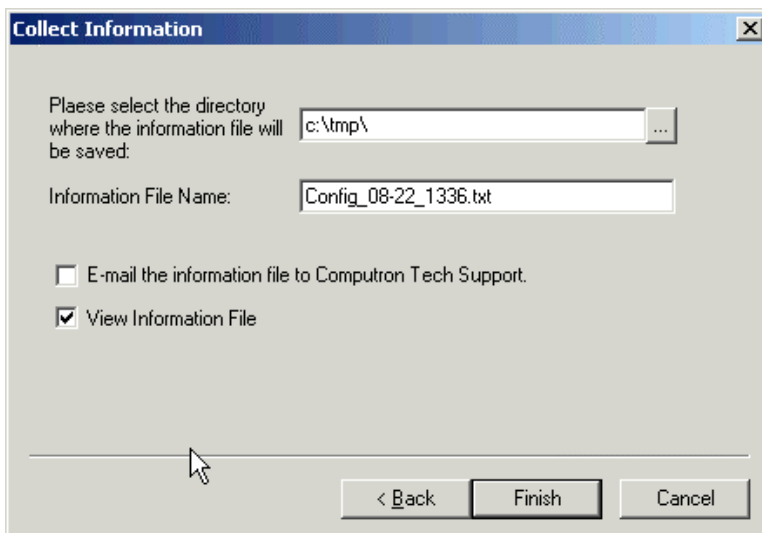
☞ *Note:* For more information on running a Program Report list, see the section titled “Program Report List”.



**Collect Information – Select application program libraries**

7. Click Next
8. Select the directory to place the “Collection Information” output in. Enter a name for the output file that is created. Place a check in the box if the file is to be emailed to Computron client support. Place a check in the box to display the report after it is generated

☞ *Note:* In order to automatically email Collect Information, it is necessary to have a MAPI compliant email client (i.e. Outlook) installed on the machine where Collect Information is being run.



**Collect Information output file**

9. Click Finish.

## 6.9 COBOL Debugging

### *Introduction to COBOL Debugging*

COBOL Debugging allows Computron personnel to work interactively with a given program or subroutine. With COBOL Debug, Computron personnel can step through the program. The changes made to enable COBOL Debug are done on a selected CSServe service.

### *Enabling COBOL Debug mode*

⚠ *Warning! Using a service with COBOL Debug enabled should only be done under the direction of Computron personnel.*

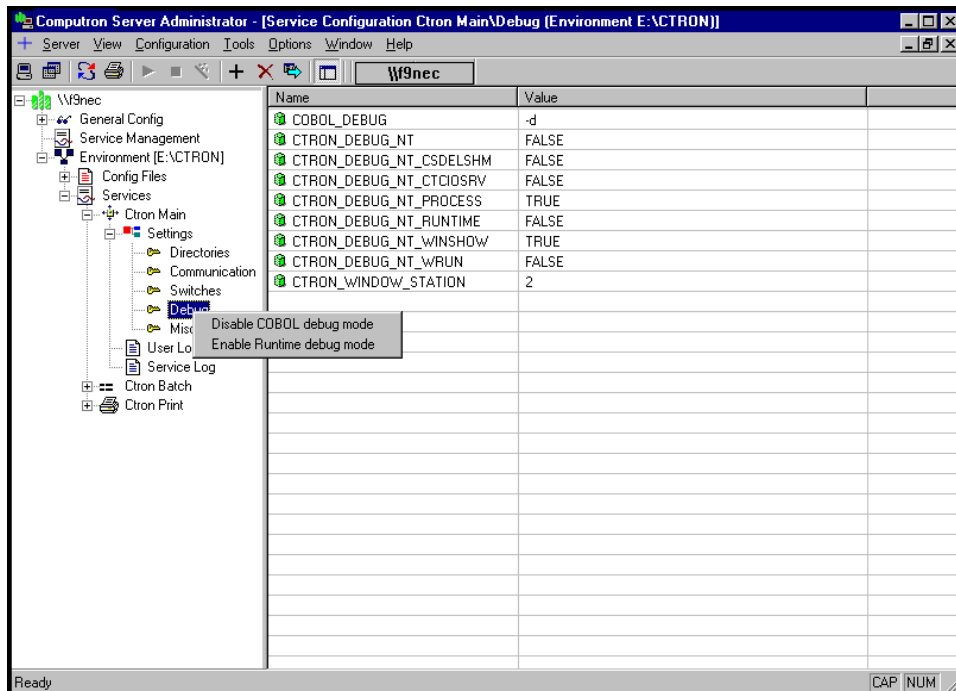
⚠ *Warning! The COBOL Debug service should never be set to start automatically. When the COBOL Debug service is not in use, it should be turned off and disabled.*

1. Create a new service by copying the main CSServe service in the environment.

☞ *Note:* For instructions on copying a CSServe service, see Section 5.10, “Installing a CSServe Service.”

⚠ *Warning! When copying an existing service it is required that a new “CTRON\_SOCKET\_NUMBER” (for VBAUI, WEBdesk, etc) and “CTRON\_TELNET\_PORT” (for Character Cell) are selected.*

2. From the service item tree in the left panel, expand (by double-clicking) Services.
3. Double-click the newly created service, to display its sub-items.
4. Double-click Settings.
5. Click on the Debug item under Settings, to highlight.
6. Right-click on the Debug item to display possible debug choices.
7. Highlight and click “Enable COBOL debug mode” from the list as shown here:

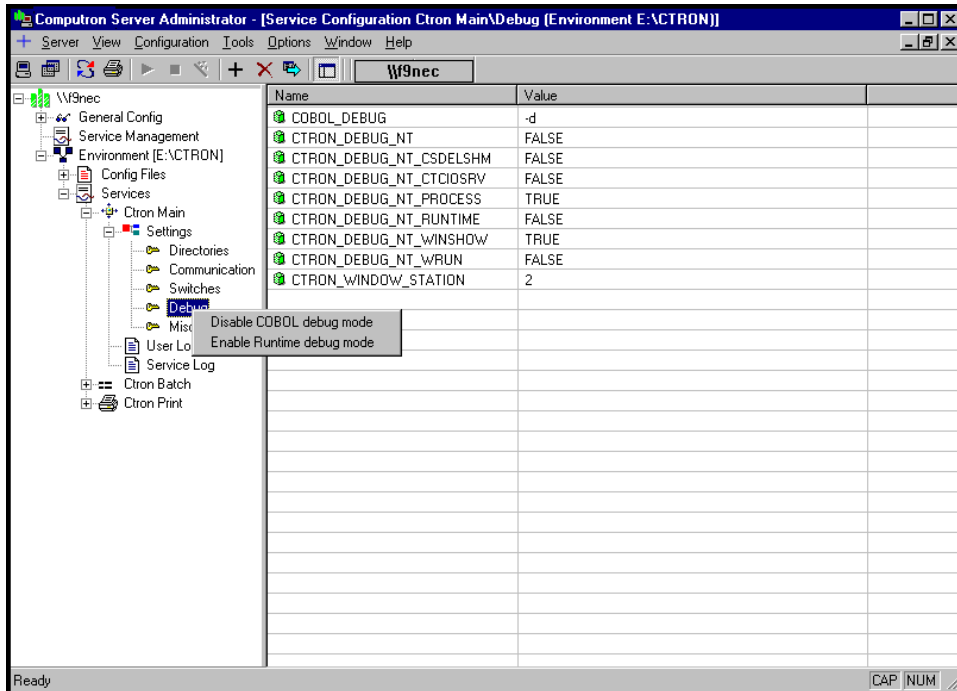


**Computron Server Administrator [Service Configuration CtronMain\Debug (Environment E:\CTRON)]**

8. This adds and modifies the entries necessary for COBOL Debug.
9. Start the COBOL Debug enabled service.

## *Disabling COBOL Debug mode*

1. Double-click the COBOL enabled service, to display its sub-items.
2. Double-click Settings.
3. Click on the Debug item under Settings, to highlight.
4. Right-click on the Debug item to display possible debug choices.
5. Highlight and click “Disable COBOL debug mode” from the list as shown here:



**Computron Server Administrator [Service Configuration CtronMain\Debug (Environment E:\CTRON)]**

6. This deletes and modifies the entries that were necessary for COBOL Debug.
7. Restart the COBOL Debug disabled service.

**Warning!** *It should not be necessary to disable COBOL debug. All debugging services should be separated, to prevent users from accessing them. Instead of disabling the COBOL debug setting, stop and disable the COBOL debug service until the next time it is needed.*

# Appendices



# Appendix A: Configuring Character Cell for Windows

## *Introduction to Configuring Character Cell for Windows*

This section details the configuration and utilization of Computron's character cell presentation for Windows clients. Character cell provides a non-graphical, text-based user interface alternative to Computron's WEBdesk web client. This feature enables sockets to Computron applications via the Telnet protocol; therefore, users with Telnet client software or VT220 terminals can connect to Computron software.

## *Architecture Overview*

The main Computron service, CSServe (csserve.exe), processes incoming Telnet requests. CSServe listens for Telnet connection requests on a socket port number that is configured in the Computron Control Panel. When a Telnet connection is established, CSServe prompts the user to enter an ID and password. The user name can be a local user (e.g., Smith) or a domain user (e.g., MyDomain\Smith). CSServe performs the exact same logon validation that is performed for web users. Thus, all access to Computron Software is controlled by native Windows security.

CSServe creates an application process on behalf of this user. Before actually running the application, this process verifies that the user is a valid Computron user by querying the Computron User file. After this final user validation and all initializations are complete, the application logic begins. All interaction with the user is directed through Computron's presentation subsystem, which detects that a character cell presentation has been requested. It sends and receives text-only versions of Computron panels over the socket created by CSServe.

Once CSServe creates the application process, it no longer plays a role in the running of that application. The application process inherits the socket created by CSServe, while CSServe returns to sleep mode and waits for a connection request (on the main port) or a Telnet connection request (on the Telnet port).

## *Software Configuration*

Each Computron configuration created with the Computron Control Panel (cssrvctl.exe) can be individually configured for character cell. On a per-configuration basis, character cell may be enabled or disabled. Character cell is enabled by specifying a unique port number to receive Telnet requests. If the port number is not defined, then character cell is disabled for that configuration.

## Enabling Character Cell

First, run the Computron Control Panel. Then, for each configuration that uses the character cell presentation, complete the following steps:

1. Highlight the configuration name, and then click Properties.

2. Click Add to add a new configuration item.
3. Specify the following:

**Name**

CTRON\_TELNET\_PORT

**Value:**

xxxx (a unique Telnet port number)

**Type:**

REG\_DWORD

4. Click OK to add the configuration item.
5. Click Save to save the configuration.
6. If this configuration is currently running, you must stop it and then restart it so that the change takes effect.

## *Running Character Cell*

The following section describes how to configure and use character cell.

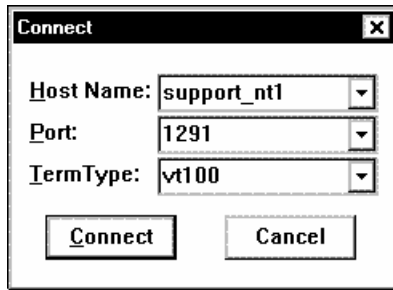
### Configuring Telnet Client Software

The main consideration for configuring Telnet client software is to enable the keyboard's arrow keys properly. If you're running the Telnet client that comes with Windows 95, this can be accomplished by clicking the Terminal-Preferences option and then selecting the VT100 Arrows check box.

### Starting Character Cell and Logging onto Computron

From a VT220 terminal that has network accessibility to the Windows Server or from Telnet client software, Telnet to the Windows Server by:

1. Start a Telnet session by either:
  - ◆ using the Start-Run command and entering Telnet in the Open box
  - ◆ double-clicking the telnet.exe file in the Windows directory.
2. From the Telnet window, click the Connect-Remote System option.



Telnet Connect Window

3. Enter the following information:

**Host Name:**

Enter the Windows Server name.

**Port Type:**

Specify a port number (unless a default of 23 was configured).

**Term Type:**

Accept the VT100 default.

4. Click Connect to open a connection to the CSServe service.
5. When prompted for a user name, enter a local user name (local to the Windows Server). To enter a domain user, specify the domain name followed by a slash (or backslash), followed by the domain user name (e.g., Mydomain\Myname).
6. Press Enter.
7. When prompted for a password, enter the user password.
8. Press Enter.

At this point, CSServe validates the user name and password. If the validation fails, the user is prompted to enter the user name and password again. After three failed attempts, the socket is closed and the user has to Telnet again in order to log on. Once a user has entered a valid user name and password, they are presented with the main Computron menu.

## Using Character Cell – Getting Started


A full discussion of the usage of character cell is beyond the scope of this document. However, a few basic pointers allow you to explore the system.

- ◆ Choose menu options by typing the menu option number; then pressing Enter.
- ◆ Pressing PF-16 (function key 16) backs up to the previous menu. Function keys are entered by pressing Ctrl+F followed by the two digit function key number, one after the other.

- ◆ When running an application, function keys (PF) are frequently used to select actions. The currently available function keys are listed on the window. Function keys that are sensitive to the cursor position are displayed in angle brackets, such as <10>. Function keys that are not sensitive to the cursor position are displayed in parentheses, such as (16).
  - PF-16 generally backs up to the previous screen.
  - PF-32 generally exits the application.
- ◆ Type a question mark (?) in a field and press Enter to invoke the Q-Mark feature, which displays a list of values to choose from.
- ◆ Use the Tab key to move from field to field or take a shortcut by using the arrow keys. Also, completely filling a field with data automatically tabs to the next field.

## Logging Out of a Character Cell Session

Exit the current application and return to a Computron menu (using PF-16 to back up to the menu). Press PF-31 to exit the menu system and close the Telnet session.

 *Tip:* Since the Telnet client stores the Host Name/Port Number information for each session, the next time you start a Telnet session you can simply select a Host Name from the most recently used list at the bottom of the Connect menu.

# 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
<b>Start</b>	Beginning of program, before display of first window.
<b>Before</b>	Just before the display of a ULM line item.
<b>Validate</b>	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
<b>Final-Val</b>	For ULM windows only, just after the final Computron process, following the use of the Process/Save option which initiates file updating.
<b>Process</b>	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.
<b>Terminate</b>	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

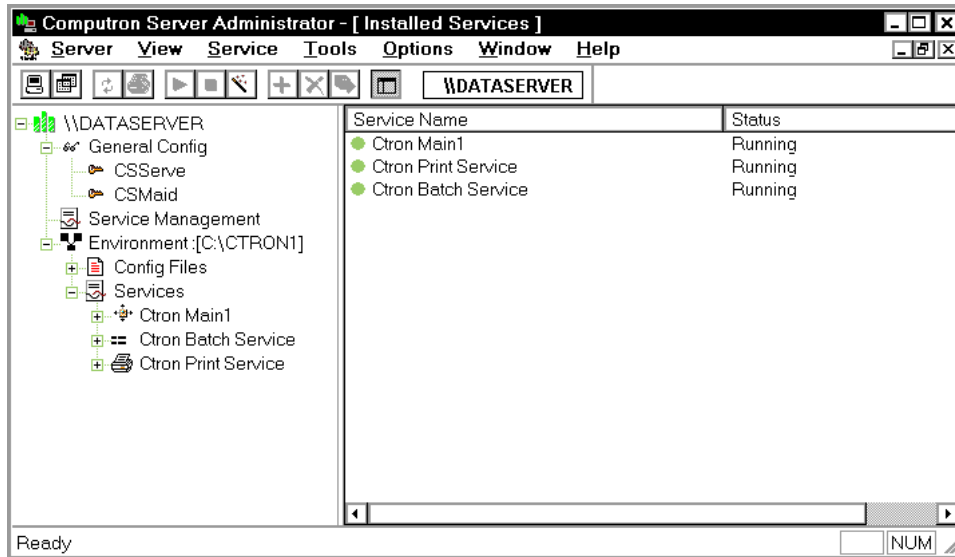
## *Implementation Overview*

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:

### **Establish the Path for Access to the Compiled UserExit**

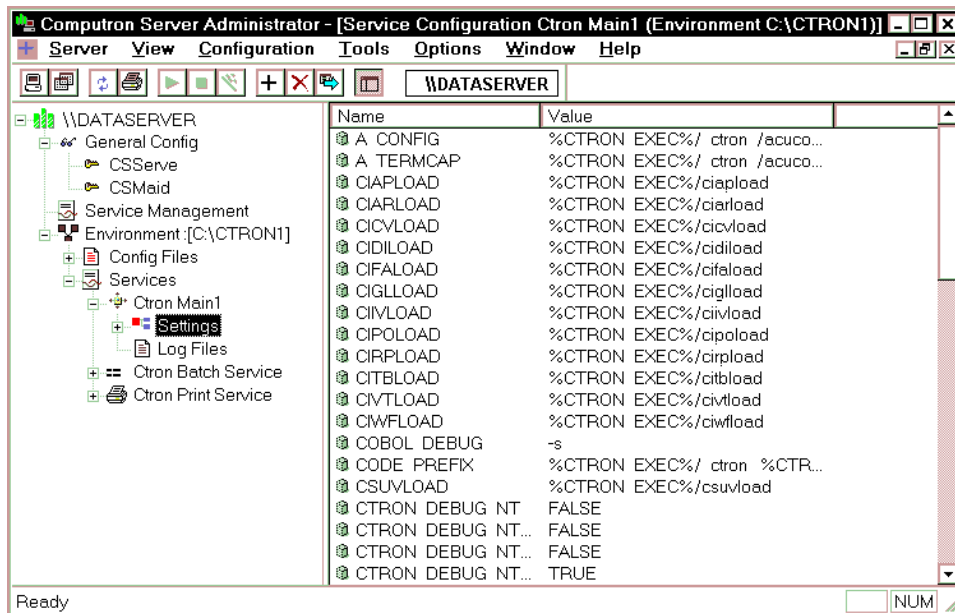
In the Windows environment, programs and subroutines for execution are located through a path of directories as defined in Computron’s CODE\_PREFIX configuration parameter. Custom file handling is set up as follows:

1. Open the Computron Server Administrator.



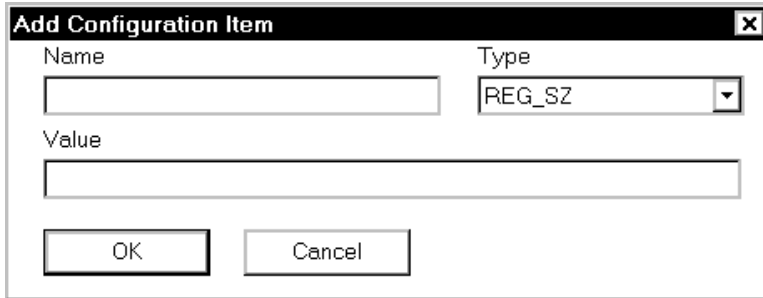
**Computron Server Administrator – [Installed Services]**

2. From the left panel, expand the Environment option to access the Computron environment where the service is installed.
3. Expand the Services option to access the services in the environment.
4. Expand the Service Name to access the Settings option.
5. Double-click the Settings option to display all the service configuration items in the right panel, as illustrated here.



**Computron Server Administrator**

6. Right-click anywhere in the right panel list to display a shortcut menu. Click the Add command to open the Add Configuration Item dialog box.



The dialog box has a title bar 'Add Configuration Item' with a close button. It contains three input fields: 'Name' (empty), 'Type' (dropdown menu showing 'REG\_SZ'), and 'Value' (empty). At the bottom are 'OK' and 'Cancel' buttons.

**Add Configuration Item**

7. Add a new configuration item, as follows:

**Name**

YPXXLOAD (where XX is the custom characters for the company)

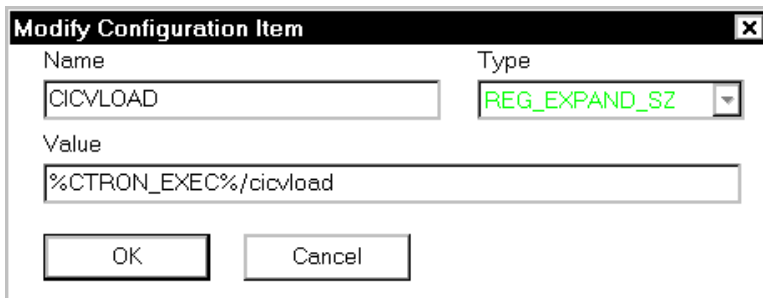
**Type**

Set this to REG\_EXPAND\_SZ.

**Value**

Set this to %CTRON\_EXEC% / ypxload

8. Click OK to add the item and return to the Server Administrator window.
9. Right-click the CODE\_PREFIX item and then click Modify from the shortcut menu to open the Modify Configuration Item dialog box.



The dialog box has a title bar 'Modify Configuration Item' with a close button. It contains three input fields: 'Name' (text box with 'CICVLOAD'), 'Type' (dropdown menu showing 'REG\_EXPAND\_SZ'), and 'Value' (text box with '%CTRON\_EXEC%/cicload'). At the bottom are 'OK' and 'Cancel' buttons.

**Modify Configuration Item**

10. From the Modify Configuration Item dialog box, edit the item as follows:

**Name**

CODE\_PREFIX

**Type**

Select REG\_EXPAND\_SZ (non-modifiable).

**Value**

Set this field to %CTRON\_EXEC% / ypxload.

(Add to the beginning of the existing value. Make sure to add a space between this entry and the rest of the path.)

11. Click OK to save the item return to the Server Administrator window.
  - ☞ *Important!* If this configuration is currently running, make sure all users are logged off the system before proceeding to the next step.
12. To implement your changes you must first stop and then restart the Windows Server, as follows:
  - a) Double-click the Service Management option in the left panel to display the service name in the right panel.
  - b) Right-click the service to display a shortcut menu.
  - c) Click the Stop command to shutdown the server. Note that the status changes from Running to Stopped.
  - d) Right-click the service again, but this time click Start on the shortcut menu.
  - e) Make sure the Status changes from Stopped to Running.

## 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 monitor) 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 monitor) 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 monitor) 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 monitor) 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 monitor) 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 is used to indicate to the system whether all the lines of an ULM window should be revalidated via the UserExit just after selecting the Process/Save option. If used, the CRT (or monitor) status is controlled by the option entered for the UserExit Validations field, above. Valid options are:

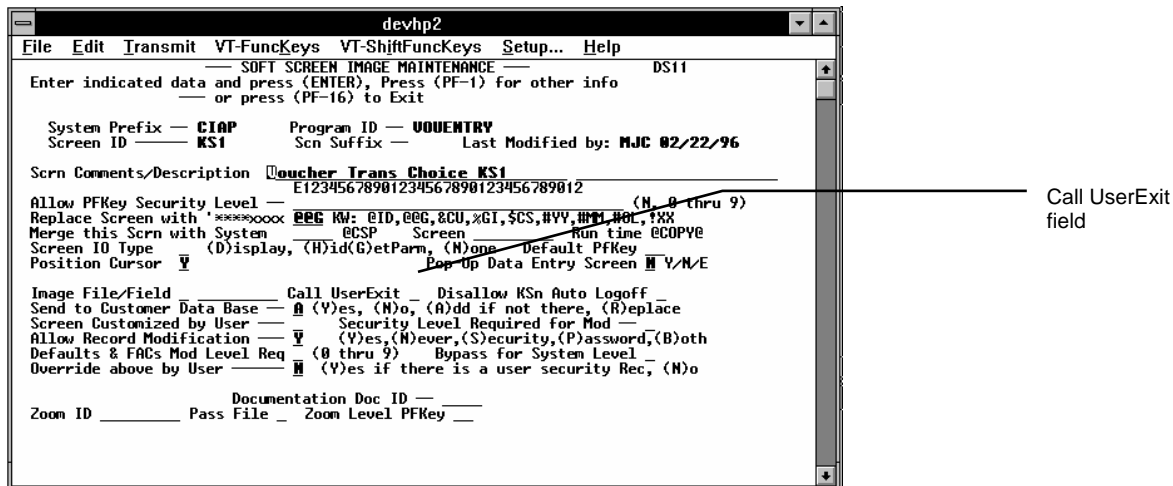
- Revalidate all the entered distribution lines after selecting the Process/Save option.
- Save the distribution lines without revalidating them.

## 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.



Soft Screens Image Maintenance Window (DS11)

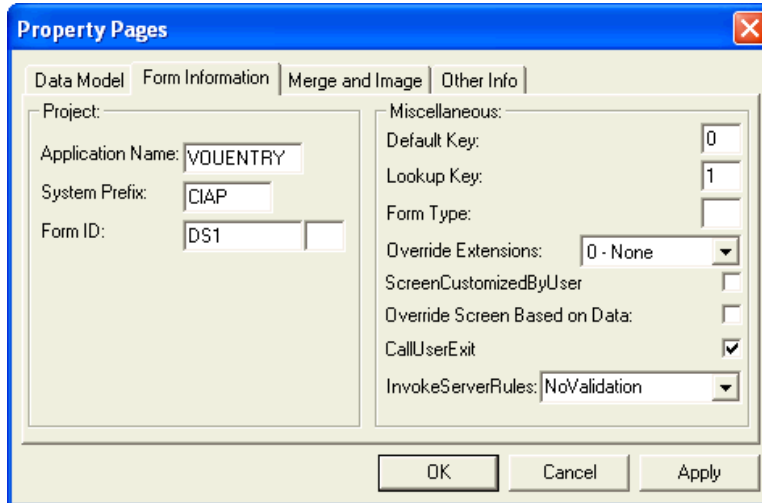
### Call UserExit

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*.

## *Implementing C UserExits with Computron Software on Windows*

### Introduction

Using Microsoft Visual C, it is possible to write Computron UserExits in C language for use with Computron Software on Windows. The UserExits are compiled into a DLL (usersub.dll) and placed in a directory on the server machine (usually ...\_ctron\_\lib). The UserExit routines are then called by the Computron runtime and the appropriate custom code is executed.

### Requirements

- ◆ Microsoft Visual C version 6 (Visual Studio) or higher or similar compiler
- ◆ “C” programming knowledge

## Building the USERSUB.DLL module

The following sections explain the usersub.dll module; it's components, and how to create it.

### *General Notes*

The source code and project files for usersub.dll are distributed in a single directory. This directory is usually `...\exec\_ctron_\lib`.

These source files should be copied to a development area on your PC and then modified, so that the original files are available if needed.

Any code changes made to the source files should be backed up and/or maintained in the usual manner particular to your site.

### *Source Files*

Below are a list of the source files and their purpose. The following files are included in the usersub.dll project that is provided:

<b>Files</b>	<b>Description</b>
USERSUB.DSP	Visual C Project file.
USERSUB.DSW	Visual C Workspace file.
USERSUB.C	Main C program – contains UserSub() routine.
USERSUB.H	Header file with macro definitions.
USERSUB.DEF	Module definition file.

The following files can be added to the project to provide access to data elements for various Computron applications:

APUEXIT.C	ARUEXIT.C	EPICWF.H	FAUEXIT.C
GLUEXIT.C	IVUEXIT.C	OI_STUB.C	POUEXIT.C
SCNGLOB.D.H	TBUEXIT.C	UECOMMON.H	UVUEXIT.C
VTUEXIT.C	WFUEXIT.C	WFUEXITD.H	

### *Creating USERSUB.DLL*

Before actually performing this step, you need to know what your UserExit is going to do and how to accomplish this with a C program.

1. Copy the contents of the `...\_ctron_\lib` directory to a development directory.
2. Run Microsoft Visual Studio and load the project by loading the workspace called `usersub.dsw` from your development directory.



- ☞ *Note:* If you are using a C compiler other than Microsoft Visual Studio, you have to create a new project for your specific environment. Make sure that your UserSub function is properly exported.
3. Write the code for your UserExit routine. Use the appropriate .C file provided as a template or create a new .C file if necessary. Use the .C file that corresponds to the application being run, i.e. gluexit.c for General Ledger, apuexit.c for Accounts Payable, etc.
  4. Add the new/modified .C file to the UserExit project.
  5. Look in the usersub.c file and un-comment the section of code that applies to the required UserExit so that it can be executed. The code you need to uncomment looks something like this:
 

```

      if (!strcmp(argv[0], "GLUEXIT")) {
          gluexit(argc, argv);
          return Okay;
      }
      
```
  - ☞ *Note:* The UserExit name passed in by the application usually corresponds to the type of application being run, i.e. GLUEXIT for General Ledger, APUEXIT for Accounts Payable, etc, so un-comment only the piece(s) of code that you need.
  6. Compile and build usersub.dll.

These are only basic steps and recommendations. They are provided here to give an understanding of the UserExit code and how it is structured.

You may choose to go beyond these steps and/or modify any of the UserExit code as necessary for your particular situation.

## Implementing USERSUB.DLL at Runtime

Now that you have properly built the UserExit module, it must be placed on the Computron server and the proper configuration entries must be added so that the module is loaded and called by the applications. Follow these steps to install the UserExit DLL:

1. Copy the usersub.dll file to the server into a directory called ...\\exec\\\_ctron\_\\lib. Create this directory if it does not already exist.
2. Run the Computron Server Administrator (ConfigMan) and open the service for which you want to add the UserExit functionality (select Services → (name of your CSServe service) in the tree).
3. Add a new configuration item to this service (open Settings in the tree and then selecting Configuration → Config Item → Add from the menu). Use the following values:

### Name

The name of the setting is CTRON\_USER\_SUB\_DLL.

### Type

Select REG\_EXPAND\_SZ.

## Value

Set the value to %CTRON\_EXEC%/\_ctron\_/lib/usersub.dll

☞ *Note:* It may be necessary to modify the path name if you have placed usersub.dll in a different directory.

4. Stop and re-start the service that you just modified.
5. Sign on to Computron Software and run the appropriate application to test the UserExit. If it seems like the UserExit is not being loaded or not executing, check the user log file (CTRON\_LOG\_userid.LOG). An entry like this indicates there is a problem locating/loading the DLL:

```
=====
04/28/07 12:48:33 PID: 00111  RUNTIME: E:\ctron\exec\_ctron_\wrunsql.exe
VER: 10.0.0.0      PGM: MENUPROG/USERSUB      QUAL: ERROR      TYPE:
Cannot load E:/ctron/exec/_ctron_/lib/usersub.dll DLL - OS error 126.
User exits will not be available.
Set CTRON_USER_SUB_DLL env variable to correct filespec.
```

The error above (OS error 126) is the most common – it indicates that the module could not be found, which means the DLL file is in the wrong place or the CTRON\_USER\_SUB\_DLL value is incorrect.

## Debugging USERSUB.DLL

There are several ways that usersub.dll can be debugged. Some methods are listed below – choose whichever debugging method is most comfortable for you.

- ◆ Add code that writes information to a log file. There is commented-out code in usersub.c which writes to a log file.
- ◆ Use message boxes. Normally, message boxes in the server code are very bad; they “display” on the server (on a virtual desktop) and there is no way to dismiss them, so the server program essentially hangs. However, you can see the message boxes on the Windows server by performing the following configuration changes:
  1. Set a config value on the CSServe service:  
  
Name: CTRON\_WINDOW\_STATION  
  
Type: REG\_DWORD  
  
Value: 2
  2. Add your Windows user to the Administrators group. This is necessary to be able to use the Computron server’s desktop.
  3. Modify your CSServe service to have the “Interact with Desktop” checkbox checked on. This can be done in Computron Server Administrator by right clicking on the service and changing its properties.
  4. Stop and re-start the CSServe service.
- ◆ Use “Attach to Process” in the Microsoft Visual Studio debugger.

- ☞ *Note:* Your user must be added to the Administrators group for this type of debugging.
- ◆ Use the Debug Break () statement in your code. This automatically breaks to the debugger.

## *Using Other Development Environments*

We recommend using Microsoft Visual Studio as your development environment to create usersub.dll.

If you decide to use another development environment or a language other than C to create this DLL, make sure that the resultant usersub.dll...

- ◆ Is a 32-bit Windows DLL
- ◆ Has an exported function called “UserSub” which takes argc/argv type arguments.

# 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 Windows 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  Path
PGMVOL  /ctron/exec
IPLVOL  /ctron/exec
EXEC     /ctron/exec
VOL777  /ctron/data
DATA    /ctron/data
```

- ◆ *Volume* can also be used to identify a Windows server or a database server. In this case, the server name is entered in lieu of the absolute path.

*Example:*

```
Volume      Path
server      ctron_database
```

(where `ctron_database` is the name defined during the ODBC setup procedure.)

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 Windows absolute pathname on the remote system can be specified.

*Example:*

```
Volume    Path
server @Host0:VOL777
serv2   @Host1:/ctron/data
```

## LGMAP Maintenance

A visual editor (like *vi* for *Windows*), 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: Batch Queue Log File Messages

## *Batch Queuing System Log File Messages*

The following examples include the most common messages that are logged by the Batch Queuing System.

Message:     **Job submitted – Job[123] User: xyz, Description :POSTIL1--ZOOM Inquiry—**

Meaning:     This message is logged when a user submits a job. The number in the brackets, 123 in this example, is the job number. Also in this example (and the ones that follow), xyz is the user ID of the user that submitted the job; the user submitted program POSTIL1; and the familiar name of that program is ZOOM Inquiry.

---

Message:     **Starting job – Job[123] User: xyz, Description :POSTIL1--ZOOM Inquiry—**

Meaning:     This message is logged when a job on the job queue begins running. The Batch Queuing System removes the job from the queue, logs this message, then runs the job.

---

Message:     **Completed job – Job[123] User: xyz, Description :POSTIL1--ZOOM Inquiry—**

Meaning:     The Batch Queuing System has completed running this job.

---

Message:     **Job[123] was held.**

Meaning:     An administrator running the Computron Process Monitor has held the job. To find out more about that job, such as who submitted it and what program it runs, you have to use the indicated job number and look for the corresponding “Job submitted” message earlier in the log file.

---

Message:     **Job[123] was released.**

Meaning:     An administrator running the Computron Process Monitor has released the job.

---

Message:     **Job[123] has been deleted from the job queue.**

Meaning:     An administrator running the Computron Process Monitor has removed (deleted) the job from the job queue. The job did not run.

---

Message:     **Job[123] job class was changed from “ ” to “B”.**

Meaning:     An administrator running the Computron Process Monitor has changed the job’s job class. In this example, the job class for job number 123 was changed from blank to “B”.

---

---

Message: **Job[123] job queue priority was changed from 5 to 6.**

Meaning: An administrator running the Computron Process Monitor has changed the job's job queue priority. Note that this only affects the order in which jobs are run (do not confuse job queue priority with run priority). In this example, job number 123's job queue priority was increased from five to six, which will cause it to run before other jobs whose priorities are five or lower.

---

Message: **Job Queue has been held.**

Meaning: An administrator running the Computron Process Monitor has held the job queue. No further background jobs will begin running until the queue is released.

---

Message: **Job Queue has been released.**

Meaning: An administrator running the Computron Process Monitor has released the job queue.

---

Message: **Starting: C:\ctron\exec\\_ctron\_\Csqserv.exe  
Version: 3.5.4.0**

Meaning: This two-line message indicates that the service is starting. The version number in this example is 3.5.4.

---

Message: **Clearing the job queue...**

Meaning: The job queue is cleared when the Batch Queuing System starts and when it ends. This message is logged when the Batch Queuing System begins clearing the queue.

---

Message: **Job[123] cleared from the queue.**

Meaning: While the job queue is being cleared, this message may be logged if the Batch Queuing System determines that a job was in the job queue at the time the service was stopped. The job listed has not run. Use the indicated job number to find the corresponding "Job submitted" message earlier in the log file in order to determine more about this job.

---

Message: **Job[123] was running (status unknown).**

Meaning: While the job queue is being cleared, this message may be logged if the Batch Queuing System determines that a job was running at the time the service was stopped. The service lost track of the job at the time the service was stopped. If the service was ended normally, then this job probably ran to completion. However, if this message was logged while starting the Batch Queuing System after a system failure, then the job probably did not complete.

---

Message: **Finished clearing the job queue.**

Meaning: After the Batch Queuing system finishes clearing the job queue (either when the service is starting or stopping), it logs this message to indicate the status. If no jobs were cleared (the job queue was already clear), then this message merely indicates that the clearing activity is complete. However, if jobs were cleared, then the following message is logged (which is similar but contains additional information).

---

---

Message: **Finished clearing the job queue.  
There were 4 jobs waiting in the job queue and 2 jobs running  
when the batch queuing service ended.**

Meaning: After the Batch Queuing system finishes clearing the job queue (either when the service is starting or stopping), it logs this message to indicate that jobs were indeed cleared from the queue. This message summarizes the preceding messages and explains the state of the Batch Queuing System at the time the service was ended (either normally or abnormally). In this example, four jobs were in the queue – these jobs did not run. Two jobs were running – their state cannot be accurately determined. See the descriptions for “Job[123] cleared from the queue” and “Job[123] was running (status unknown)” for more information.

---

Message: **The Batch Queuing System is ready.**

Meaning: The service has completed its initializations and is ready to accept and process jobs.

---

Message: **Shutting down: C:\ctron\exec\\_ctron\_\Csqserv.exe**

Meaning: The Batch Queuing System is being stopped. It will stop accepting new jobs and clear the existing jobs from the queue. After this cleanup activity is complete, it will log the following message.

---

Message: **The Batch Queuing System has stopped.**

Meaning: The service was requested to stop. The service has completed its cleanup activity. An instant after logging this message, the service process ends.

---



# Appendix E Backup and Recovery

This section attempts to outline basic backup and recovery procedures for Computron Software running under the Windows operating system. In addition to these recommendations, Computron pledges to work with each client to assist in defining procedures for maintaining your mission critical server as it relates to Computron's application.

Keep in mind, however, that specific details of Windows and MS SQL administration are the responsibilities of individual clients. Due to the fact that each client network/system configuration can vary, Computron recommends that clients have certified Microsoft and SQL administrators to define the specific details entailed to fit within their administrative scheme.

## *Initial Backup System Setup*

When creating a backup Windows Server system, follow these guidelines:

- ◆ If you are using a second machine as a backup Windows Server, make sure that it has enough disk storage and/or memory capacity to handle current *and* future production volumes. Computron highly recommends monitoring this capacity on an ongoing basis.
- ◆ Install and configure a fully functional system of Computron Software on your backup WINDOWS Server (including necessary 3rd party software, like MKS, ODBC, MSSQL, etc.).
- *Warning! Use caution when copying your Computron Production file system to the backup server — especially if the file system names are not identical (e.g., your Production system is named CtronProd and your backup system is CtronTest). Otherwise, you may inadvertently overwrite unique entries in the LGMAP file on the backup server.*
- ◆ Install and configure Computron applications. Part of the Computron's configurations are the registries, services, and the NTStart utility.
  - ☞ *Note:* These settings are integrated and maintained as part of the Windows database. Therefore, each entry is factored with existing registry entries, as well as the order of creation/updates. Once these registries are set up by the installation procedure, Computron recommends performing parallel changes on each respective server.
- ◆ Replicate the existing Windows user accounts — if your users are defined on a local server. If your users are defined on a Windows domain, then the server needs to be part of the domain. (See your Windows System Administrator for specifics.)
- ◆ Create databases on the backup server that are *identical* to the ones on the production server that you are restoring to.
- ◆ In a networked Windows domain environment, you must establish access to the backup environments by making one of the following adjustments:

- change the domain server name, so that the test server environment mimics the production server environment;
- change the IP address of the test server to match the production address;
- create another Host Entity for the test environment in Computron's CSCOMM.INI file (via the **Setup** option on the Computron Login window).

## Initial Certification of the Backup Environment

After establishing your backup server environment it is extremely important that you thoroughly test its overall integrity *before* a disaster occurs. Your goal should be to ensure that the backup environment is an *exact* replica of the production environment, using the following checklist as a rough guideline:

- ◆ make sure that all security settings are in place;
- ◆ verify all software customizations, including:
  - menus
  - screens
  - programs
  - reports
- ◆ perform a comprehensive test of all application functions and reports.

## Scheduled Backup Process

Once the “Initial Setup” steps have been completed, you should make the following tasks part of your backup routine.

### Database Backups

☞ *Note:* These files can be backed up to tape or to a designated backup database server. If tapes are used though, then a restore process must be defined.

#### *Full Backup*

You could use a daily *full-backup* method if storage and/or memory constraints are *not* an issue:

- ◆ daily full backup of the MSSQL databases (using native backup tools)
- ◆ daily backup of your user files:
  - operating system user files (Windows SAM user accounts)
  - Computron User file (USERFIL3) — typically, \Ctron\Exec\\_Ctron\_

#### *Incremental Backup*

You could use an incremental backup method if storage and/or memory constraints are an issue, *and* if you are willing to use the *Roll Forward* procedure to retrieve your data:

- ◆ weekly backup the Computron data directories — typically, \Ctron\Data
- ◆ daily backup of the MSSQL database's daily transaction logs
- ◆ daily backup of your user files:
  - operating system user files (Windows SAM user accounts)
  - Computron User file (USERFIL3) — typically, \Ctron\Exec\\_Ctron\_

## Executable Program File Backups

☞ *Note:* These files can be backed up to tape or to a designated backup database server. If tapes are used, then a restore process must be defined.

- ◆ a monthly or periodic backup (usually upon a software update) of Computron's executable directories (typically \Ctron\Exec) is usually sufficient.

## Incremental Restore Process

These files would be used in the event that data needs to be restored – including transactions processed since the last backup:

- ◆ backup of the MSSQL database's daily transaction logs;
- ◆ if restoring from a tape backup, use the latest versions of the \Ctron\Exec and \Ctron\Data directories.

Computron recommends that any other changes made to the Production server (e.g., registries, SAM, MSSQL, etc.) should be added/changed in parallel on their respective servers.

## Certifying the Restore Process

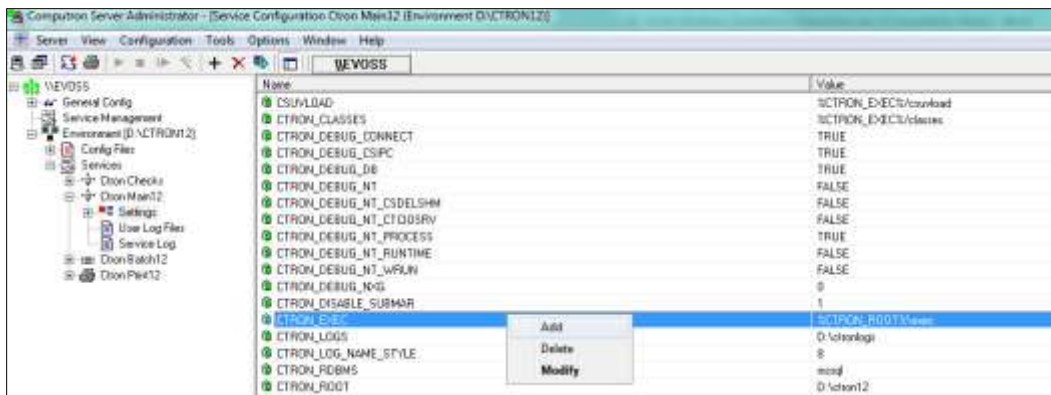
To certify that your restore process is in place, Computron recommends the following minimum requirements:

- ◆ test mission-critical functions and reports used during daily processing;
- ◆ make sure that you can access all of your data (e.g., using Zoom Inquiry);
- ◆ verify that your global variables are intact;
- ◆ verify that the most recently completed transaction before the restore process took place was completed.

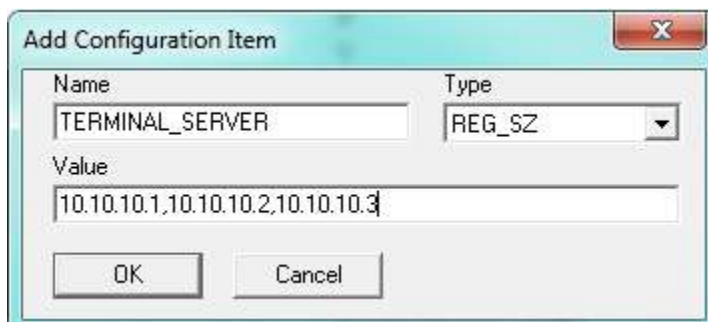
# Appendix F – Configuration Change to a Support Terminal or Web Server

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 iterative connections from the same IP address.

To implement this change, run the Computron Server Administrator (aka ConfigMan) and locate the **Main Service** in the tree on the left panel. Expand the **Settings** key, as shown below:



In the right panel, right-click and select Add. In the ensuing dialog, add a switch called **TERMINAL\_SERVER**, of type **REG\_SZ**. In the Value field, add the IP address(es) of each Terminal Server and/or WEBdesk/e-Cellerator Web Server which will supply client connections to this application server. Delimit the IP addresses with commas and do not include any spaces. See the example below:



Click OK when you have finished. Now, in ConfigMan's left panel, click on the Service Management key in the tree. Next, in the right panel, right-click on the Main Service and select Restart. When executed, the service stops and immediately starts. This step is necessary so that the new switch is added to the active configuration.

**Note:** In order to restart a Main Service, it is necessary to have all users logged off this server and you must ensure that all runtime (wrunsql.exe or wrunora.exe) processes have been terminated. Check Task Manager to verify and terminate any remaining runtimes after all users have exited the system.