

**Retail Application Interface Suite for AS/400
V4R3M0 5X99RAI**

Take Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page xiii.

First Edition (March 2000)

This edition applies to the licensed programs IBM Operating System/400 (Program 5769-SS1), Version 4 Release 3 Modification 0; and IBM Retail Application Interface Suite for AS/400 (Program 5X99RAI), Version 4 Release 3 Modification 0, and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the proper edition for the level of the product.

Order publications through the Rochester IBM Custom Technology Center. Publications are not stocked at the address given below.

A form for reader comments is provided at the back of this publication. If the form has been removed, you can mail your comments to:

IBM Rochester
5X99RAI Development - Department G2V
3605 Highway 52 N
Rochester, MN 55901-9986 USA

or you can fax your comments to:

(+1)+507+253-5156

If you have access to Internet, you can check the Custom Technology Center web page at: http://www.as400.ibm.com/Service/welcome_3.htm

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you or restricting your use of it.

Copyright International Business Machines Corporation 2000. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	xiii
Programming Interface Information	xiii
Trademarks and Service Marks	xiii
About Retail Application Interface Suite for AS/400	xv
Who Should Use This Book	xv
Chapter 1. Retail Application Interface Suite for AS/400 Planning and Installation	1-1
Functions Provided	1-1
Retail Communications API	1-1
Advanced Data Communication for Stores (ADCS) Emulation	1-1
Host Command Processor Emulation (HCP Emulation)	1-2
Retail File Conversion	1-2
Retail File Polling Application	1-2
Relational database collection and template support	1-2
Retail Application Logical Interface definitions for the GSA relational database	1-3
Point-of-Sale Environments	1-3
AS/400 System as an In-Store Processor	1-3
AS/400 System as Host System for a Point-of-Sale Network	1-5
Planning for Retail Communications	1-6
Planning Requirements	1-6
Installation Information	1-7
Installing the Utility - LODRUN	1-8
Installing the Utility options	1-8
Adding the RAIS License Key	1-9
Saving the Utility	1-9
Deleting the Utility	1-10
Retail Application Interface Suite effect on customer applications	1-10
Retail Application Interface Suite integration with customer applications	1-10
Retail Application Interface Suite location manager concerns	1-10
Retail Application Interface Suite coexistence with the Point-of-Sale Utility (5769CF1)	1-11
Chapter 2. Getting Started with the Retail Application Interface Suite Base Option	2-1
Getting Started with the Retail Application Interface Suite	2-1
Configuring for Retail Communications	2-1
Configure Retail Communications Application Menu	2-1
Work with Retail Store Information	2-2
Add Retail Store Information	2-2
Change Retail Store Information	2-3
Confirm Delete of Retail Store Information	2-3
Display Retail Store Information	2-3
Rename Retail Store Information	2-4
Working with Retail Communications	2-4
Retail File Communications Configuration	2-4
Using Retail Connectivity Option server	2-5
ADXSRVTL.286 - 4690 TCP/IP Server	2-5
ADXBFTBL.286 - 4690 File Table Maintenance Program	2-6
File Structures	2-7
Keyed Files on a Retail Controller	2-7
Sequential Files on a Retail Controller	2-7
Retail File Conversion	2-7
Work with Retail Conversion Templates	2-7
Work with Records in Retail Template	2-11
Work with Fields in Template Record	2-14
Steps to Create a Retail Conversion Template	2-20
Steps to Create a Retail Database Collection Conversion Template	2-21
Retail File Conversion Considerations	2-22

Data Conversion Routine	2-23
Retail Conversion Templates	2-23
User Conversion Requirements	2-23
Point-of-Sale Template Records and Fields	2-23
Point-of-Sale Data Fields Conversion	2-23
Point-of-Sale Conversion High-Level Source Creation	2-23
Special Handling for Data Type 7=Item Code	2-24
Create Retail Source (CRTRTL SRC) Command	2-25
Print Retail Conversion Template (PRRTLPL) Command	2-26
Work with Retail Templates (WRKRTLPL) Command	2-26
Chapter 3. Retail File Application Programming Interfaces	3-1
ADCS Emulation Application Program Interface (API)	3-1
RAIS Communications Application Program Interface (API)	3-4
Using the RAIS API	3-4
RAIS API file functions	3-4
RAIS API Environment Variables	3-4
RAIS API location function	3-5
RAIS API file naming	3-5
Using ILE COBOL/RPG Program to Call APIs	3-6
RAIS Communications API Reference	3-7
Chapter 4. Point of Sale/2000	4-1
Using Retail File Communications	4-1
Retail File Communications Function	4-1
AS/400 Retail Application Interface Suite (RAISCTL) Menu	4-1
Work with Retail Communications	4-1
ADCS Emulation Point-of-Sale File Types Supported	4-2
Retail Connectivity Option Point-of-Sale File Types Supported	4-3
Retail File Communications Control Language Support	4-3
Receive Retail File (RCVRTLF) Command	4-3
Send Retail File (SNDRTLF) Command	4-7
Clear Retail File (CLRRTLF) Command	4-10
Create Retail File (CRTRTL F) Command	4-12
Delete Retail File (DLRTL F) Command	4-15
Send Retail Record (SNDRTL RCD) Command	4-17
Submit Retail Program (SBMRTL PGM) Command	4-19
Delete Retail Record (DLRTL RCD) Command	4-21
Using Host Command Processor (HCP) Emulation	4-24
HCP Commands Supported	4-24
Work with Host File Cross References	4-24
Add Host File Cross Reference	4-25
Change Host File Cross Reference	4-25
Confirm Delete of Host File Cross References	4-25
Rename Host File Cross Reference	4-25
HCP Emulation Considerations	4-26
Host File Name Cross-Reference List	4-26
HCP Emulation Message Queue	4-26
Retail User Profile Used by HCP Emulation	4-26
Steps Required to Use HCP Emulation	4-26
HCP Sense Codes Returned to the Host System	4-27
Using Retail File Conversion Displays	4-27
Retail Application Interface Suite for AS/400 Main Menu	4-27
Work with Retail File Conversion	4-27
Convert from Retail File Format	4-28
Conversion Commands	4-30
Chapter 5. RAIS Retail Polling Application	5-1
Work with Retail Polling Entries	5-1
Polling Application Commands	5-4

Retail Polling Application Notes	5-5
Retail File Table Configuration	5-5
Retail Application File Considerations	5-7
Polling Application Exit Program Considerations	5-7
Polling Application Job Description	5-7
Polling Application Performance Considerations	5-7
Polling Application Authority Considerations	5-8
Polling Application Status Via Program Call	5-8
Chapter 6. Retail Application Logical Interface	6-1
RALI Relational database considerations	6-1
Populating Optional Relational database.	6-1
RALI Views	6-1
TRANAMOUNT - Transaction Amount Table	6-3
HEADER - Header Key Table	6-4
MDSETOTALS - Transaction Merchndise Totals Amount	6-5
TAXAMT - Transaction Tax	6-6
TAXCODE - Transaction Tax Tax Code	6-7
DISCOUNT - Transaction Discounts	6-8
TENDER - Transaction Tender Amounts	6-9
TENDERCHG - Amount of Tender Change	6-10
TRANSVOID - Sales on Void Transaction	6-11
POSTVOID - Sales on Post Void Tranactions	6-12
DISCREPANCY - Transaction Discrepancy Table	6-13
MDSESALES - Merchandise Sales Table	6-15
LAYMDSE - Layaway Merchandise Sales Table	6-16
ITEMDISCOUNT - Item Discounts Given	6-17
ITEMDEPOSIT - Item Deposits	6-18
ITEMVOID - Sales on Voided Items	6-19
VPLIVOID - Sales of Line Item Voids	6-20
ISSUETNDR - Gift Certificate and Store Credit Issued	6-21
ACCTNEW - Total Amount of New/Adjusted Accounts	6-22
ACCTPAY - Total Amount of Account Payments	6-23
ACCTDEP - Total Amount of Account Deposits	6-24
ACCTFEE - Total Amount of Fees on Accounts	6-25
LAYCAN - Total Amount of Cancels on Layaways	6-26
LAYRSTK - Total Amount of Restocks on Layaways	6-27
PICKUPS - Transaction Pickup Amounts	6-28
PAYOUTS - Transaction Payout Amounts	6-29
PAYINS - Transaction Payin Amounts	6-30
TNDRCASH - Tender Cashed Amounts	6-31
SPRSNSALES - Total Amount of Salesperson Sales	6-32
SPECIALSALES - Special Sales Amounts	6-33
Chapter 7. RAIS Applications/Reports	7-1
Application/Report Command Definitions	7-1
Remove Retail Transactions (RMVRTLNS) Command	7-1
Print Retail Report (PRTRTLRPT) Command	7-2
Report Definitions	7-3
Hourly Totals Report	7-3
Cumulative Item Sales Totals Report	7-3
Item Sales Hourly Count Report	7-4
Transactions Hourly Count Report	7-4
Cashier/Register Hourly Totals Reports	7-4
Hourly Productivity Report	7-5
Salesperson Totals Report	7-5
Salesperson Department Totals Report	7-6
Appendix A. Problem Analysis	A-1
Detecting the Problem	A-1

Re-creating the Problem	A-1
Gathering Problem Analysis Information	A-1
Communicating the Problem	A-1
Appendix B. Examples	B-1
RAIS API Example	B-1
Retail Controller Configuration Examples	B-1
Nonswitched Retail Controllers Configuration	B-1
Switched Line Retail Controllers Configuration	B-2
X.25 Retail Network Configuration	B-2
TCP/IP Configuration for Communication to 4690 RCO	B-2
Appendix C. Utility Trailer Record	C-1
Appendix D. QRTXHASH Randomizing Routine	D-1
Return Codes	D-1
Appendix E. AS/400 System Error Codes	E-1
AS/400 UNIX Error Codes	E-1
Appendix F. General Sales Application Cross Reference	F-1
EALTRANS - GSA Transaction Summary Log	F-2
String '00' Transaction Log Header to QAZCFGHDR	F-3
String '00' Transaction Log Header to QAZCFGTTIM	F-5
String '01' Header String Extension to QAZCFGHDRX	F-6
String '02' Price Change to QAZCFGPRCH	F-6
String '03' Totals String to QAZCFGTOTL	F-7
String '04' Cash Drawer Change String to QAZCFGCSDW	F-8
String '05' Store Close to QAZCFGCLOS	F-8
String '06' Merchandise / Non Merchandise String to QAZCFGMRCH	F-9
String '07' Merchandise / Non Merchandise String Extension to QAZCFGMRCX	F-11
String '08' Miscellaneous String to QAZCFGMISC	F-12
String '09' Data Entry to QAZCFGDTAE	F-13
String '10' Discount to QAZCFGDISC	F-14
String '11' Cash Tender String to QAZCFGTNRD	F-14
String '12' Cash-Like Tender String to QAZCFGTNRD	F-15
String '13' Tender Extension String to QAZCFGTNRDX	F-16
String '14' Tax String to QAZCFGTAX	F-17
String '15' User Data String to QAZCFGUSER	F-17
String '15' User data	F-17
String '16' Log Time Analysis String to QAZCFGTTIM	F-18
String '17' Void Previous Transaction String to QAZCFGVOID	F-19
String '18' Data Maintenance to QAZCFGDTAM	F-19
String '19' Invalid Sign-on String to QAZCFGSIGN	F-20
String '20' Modify Ticket to QAZCFGMDFY	F-20
String '23' Multiple Currency to QAZCFGMCF	F-21
String '25' Full Screen Customer information to QAZCFGRCPT	F-22
String '26' Full Screen Send Delivery to QAZCFGSNDL	F-23
String '27' Tender Removal to QAZCFGTNRD	F-24
String '28' Void Coupons String to QAZCFGTAX	F-25
Appendix G. Supermarket Application Cross Reference	G-1
EAMTRAN* - SA Transaction Summary Log	G-2
String '00' Transaction Log Header to QAZCFSHDR	G-3
String '00' Transaction Log Header to QAZCFSTTIM	G-5
String '01' Merchandise / Non Merchandise String to QAZCFSMRCH	G-6
String '01' Merchandise / Non Merchandise String to QAZCFSTNRD	G-8
String '02' Merchandise / Non Merchandise String Extension to QAZCFSMRCX	G-9
String '03' and '04' Discount/void Discount String to QAZCFSDISC	G-10
String '05' and '06' Tender/Tender Correction String to QAZCFSTNRD	G-11

String '07' and '08' Tax/Tax Refund String to QAZCFSTAX	G-12
String '09' Food Stamp Change String to QAZCFSTNDR	G-13
String '10' Manager Override String to QAZCFSMGRO	G-14
String '11' Transaction Data Entry String to QAZCFSDTAE	G-15
String '12' Transaction Price Change String to QAZCFSPRCH	G-16
String '13' Transaction Cash Drawer String to QAZCFSCSDW	G-17
String '14' Transaction Item Record Change String to QAZCF SIRCH	G-18
String '15' Transaction Trading Stamps String to QAZCF SSTMP	G-19
String '20' Transaction Exception Log String	G-20
String '21' Store Closing String to QAZCF SHDR	G-20
String '21' Store Closing String to QAZCF SCLOS	G-20
String '98' Operator in Training String to QAZCF SOTRN	G-22
String '99' User Data String to QAZCF SUSER	G-22
Appendix H. Transaction Detail Type Definitions	H-1
Valid Types of Tenders	H-1
Valid Types of Transaction Functions	H-3
Types of Data Entry	H-7
EFT - Valid EFT Actions	H-8
Appendix I. Authorization	I-1
Brief Security Overview	I-1
QRAIS User Profile Authorizations	I-1
Data Access User Authorizations	I-1
Granting authorizations	I-1
Provided Authorization Lists	I-1
Granting for User Profile	I-1
Granting for Authorization List	I-1
Using QRAIS as a group profile	I-2
Bibliography	I-1
AS/400 Books	I-1
Retail Host Books	I-1
Retail Controller Books	I-1
Index	X-1

Figures

1-1. AS/400 Retail Systems In-Store Processor Environment	1-4
1-2. AS/400 Retail Systems Host Environment	1-5
3-1. COBOL/400 API Coding Example	3-1
3-2. RPG/400 API Coding Example	3-1

Tables

2-1. Valid AS/400 and Retail Data Type Combinations	2-18
4-1. Retail File Communication Commands	4-3
4-2. RCVRTLF Command Reference Matrix	4-7
4-3. Format of Data in Error Log	4-35
5-1. Format of User Exit Parameters	5-7
5-2. Format of API Data in Error Log	5-8
6-1. TRANHEADER - Transaction Header Table	6-3

6-2.	HEADER - Header Key Table	6-4
6-3.	MDSETOTALS - Transaction Merchandise Totals Amount	6-5
6-4.	TAXAMT - Amount of Tax on Sold Items	6-6
6-5.	TAXCODE - Transaction Tax Tax Code	6-7
6-6.	DISCOUNT - Transaction Discounts	6-8
6-7.	TENDER - Transaction tender amounts	6-9
6-8.	TENDERCHG - Amount of tender change	6-10
6-9.	TRANSVOID - Amount of Sales Lost to Cancels	6-11
6-10.	POSTVOID - Amount of Sales Lost to Post Voids	6-12
6-11.	DISCREPANCY - Transaction Discrepancy Table	6-14
6-12.	MDSESALES - Merchandise Sales Table	6-15
6-13.	LAYMDSE - Layaway Merchandise Sales Table	6-16
6-14.	ITEMDISCOUNT - Item Discounts Given	6-17
6-15.	ITEMDEPOSIT - Item Deposits	6-18
6-16.	ITEMVOID - Amount of Sales Lost to Item Voids	6-19
6-17.	TVPLVOID - Amount of Sales Lost to Line Item Voids	6-20
6-18.	ISSUETNDR - Gift Certificate and Store Credit Issued	6-21
6-19.	ACCTNEW - Total Amount of New/Adjusted Accounts	6-22
6-20.	ACCTPAY - Total Amount of Account Payments	6-23
6-21.	ACCTDEP - Total Amount of Account Deposits	6-24
6-22.	ACCTFEE - Total Amount Account Fees	6-25
6-23.	LAYCAN - Total Amount of Cancels on Layaways	6-26
6-24.	LAYRSTK - Total Amount of Restocks on Layaways	6-27
6-25.	PICKUPS - Transaction pickup amounts	6-28
6-26.	PAYOUTS - Transaction payout amounts	6-29
6-27.	PAYINS - Transaction payin amounts	6-30
6-28.	TNDRCASH - Tender cashed amounts	6-31
6-29.	SPRNSALES - Total Amount of Salesperson Sales	6-32
6-30.	SPECIALSALES - Special Sales Made	6-33
F-1.	String Description	F-1
F-2.	String '00' Transaction Log Header	F-3
F-3.	String '00' Transaction Log Header	F-5
F-4.	String '01' Header String Extension (Re-entry)	F-6
F-5.	String '02' Price Change	F-6
F-6.	String '03' Totals String	F-7
F-7.	String '04' Cash Drawer Change String	F-8
F-8.	String '05' Store close	F-8
F-9.	String '06' Merchandise / Non Merchandise String	F-9
F-10.	String '07' Merchandise / Non Merchandise String Extension	F-11
F-11.	String '08' Miscellaneous String (For Tax Codes)	F-12
F-12.	String '09' Data Entry	F-13
F-13.	String '10' Discount	F-14
F-14.	String '11' Cash Tender String	F-14
F-15.	String '12'Cash-Like Tender String	F-15
F-16.	String '13'Tender Extension String	F-16
F-17.	String '14' Tax String	F-17
F-18.	String '15' User data	F-17
F-19.	String '16' Log Time Analysis String	F-18
F-20.	String '17' Void Previous Transaction String	F-19
F-21.	String '18' Data Maintenance	F-19
F-22.	String '19' Invalid Sign-on String	F-20
F-23.	String '20' Modify Ticket	F-20
F-24.	String '23' Multiple Currency	F-21
F-25.	String '25' Full Screen customer information	F-22
F-26.	String '26' Full Screen Send Delivery	F-23
F-27.	String '27' Tender Removal	F-24
F-28.	String '28' Void Coupons String	F-25
G-1.	String Description	G-1
G-2.	String '00' Transaction Log Header	G-3
G-3.	String '00' Transaction Log Header	G-5

G-4.	String '01' Merchandise / Non Merchandise String	G-6
G-5.	String '01' Manufacturer/Store Cupon String	G-8
G-6.	String '02' Merchandise / Non Merchandise String Extension	G-9
G-7.	String '03' and '04 Discount/ Void Discount String	G-10
G-8.	String '05' and '06 Tender/ Tender Correction String	G-11
G-9.	String '07' and '08 Tax/ Tax Refund String	G-12
G-10.	String '09' Food Stamp Change Correction String	G-13
G-11.	String '09' Food Stamp Change Correction String	G-14
G-12.	String '11' Transaction Data Entry String	G-15
G-13.	String '12' Transaction Price Change String	G-16
G-14.	String '13' Transaction Cash Drawer Entry String	G-17
G-15.	String '14' Item Record Change String	G-18
G-16.	String '15' Transaction Trading Stamps String	G-19
G-17.	String '21' Store Closing String	G-20
G-18.	String '21' Store Close String	G-20
G-19.	String '98' Operator in Training String	G-22
G-20.	String '99' User Data String	G-22
H-1.	Valid Types of Tenders	H-1
H-2.	Valid Types of Transaction Functions	H-3
H-3.	Valid transaction data entry types	H-7
H-4.	Valid EFT Actions	H-8

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594, U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact the software interoperability coordinator. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Address your questions to:

IBM Rochester
5X99RAI Development - Department G2V
3605 Highway 52 N
Rochester, MN 55901-9986 USA

This publication could contain technical inaccuracies or typographical errors.

This publication may refer to products that are announced but not currently available in your country. This publication may also refer to products that have not been announced in your country. IBM makes no commitment to make available any unannounced products referred to herein. The final decision to announce any product is based on IBM's business and technical judgment.

Changes or additions to the text are indicated by a vertical line (|) to the left of the change or addition.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

This publication contains small programs that are furnished by IBM as simple examples to provide an illustration. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. All programs contained herein are provided to you "AS IS". THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.

Programming Interface Information

The *Retail Application Interface Suite for AS/400* book describes the function provided by the utility to communicate with retail controllers. It contains examples for using the functions of the utility. Unless specifically stated otherwise, the information in this guide must not be used for programming purposes.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

Application System/400
APPN
AS/400
CICS
FORTRAN/400
IBM
ILE COBOL/400

ILE RPG/400
IMS
NetView
Operating System/400
OS/400
System/370
400

About Retail Application Interface Suite for AS/400

This book describes and provides examples for using the functions provided by the Retail Application Interface Suite for AS/400 for support of the IBM 4680, 4684, 3651, and 3684 store systems. The book is used to enable the AS/400 host system or in-store processor to communicate with retail controllers. For a list of related publications, see the "Bibliography" on page I-1 .

Who Should Use This Book

This book is intended to be used by:

- An applications programmer familiar with communications programming for the AS/400 system

- A host applications programmer familiar with host communications, Advanced Data Communications for Stores (ADCS) programming, and store systems and applications

- A system operator who has had formal AS/400 system training, is familiar with configuring and operating the AS/400 system, and is familiar with ADCS functions

- A user who is performing problem analysis

IBM Programming Service personnel who are responsible for resolving noncustomer problems with system programs and this communication application

Before you use this book, you should be knowledgeable in the following:

- Advanced Data Communication for Stores (ADCS) functions

- Type of retail controller that you will be using

- AS/400 communications

- AS/400 system operation and functions

Throughout this book, point-of-sale controller, or store controller is referred to as retail controller.

This book makes reference to several retail controller models. For purposes of convenience, the 4681, 4684, and 4692 retail controllers should be included wherever the 4680 retail controller is referenced.

Chapter 1. Retail Application Interface Suite for AS/400 Planning and Installation

This chapter introduces Retail Application Interface Suite for AS/400 and describes the suite's purpose, the functions provided, and how to plan for and install the application suite.

The Retail Application Interface Suite for AS/400 (RAIS) is a suite of applications for communication with the IBM* 4690, 4684, 3651, and 3684 store systems. Although it is intended to work with the IBM Store systems applications, the RAIS application programming interface extends the suite to include non-IBM store systems.

The IBM 4690, 4684, 3651, and 3684 store systems are designed to be used by the retail, drug, and grocery industries. With IBM store systems, retailers are able to augment traditional cashiering and checkout functions with a variety of management functions such as sales tracking, accounting, maintaining price files, electronic funds transfer, and credit authorization.

Companies that use IBM store systems typically have a host system at a remote site and a network of retail controllers and terminals in their stores. The AS/400* system with the Retail Application Interface Suite can be installed in a store network as:

- An in-store processor between a host and retail controllers
- A host system to one or more remotely located retail controllers

Functions Provided

The functions provided by the Retail Application Interface Suite consist of:

- Retail communications application programming interface
- Advanced Data Communication for Stores (ADCS) I/O emulation
- Host command processor (HCP) emulation
- Retail file conversion
- Retail file trickle polling application
- General Sales Application (GSA) relational database collection definition and template support definition
- Retail Application Logical Interface definitions for the GSA relational database collection

The functions that you use depend on whether the AS/400 system is operating as an in-store processor or a host system. The Retail Application Interface Suite has both an

interactive menu interface and a control language interface for unattended operation. This interface enables a user-written program to access specified functions of this application suite.

The Retail Application Interface Suite supports IBM Programmable Store Systems (PSS) retail controllers (3651, 3684, 4690, and 4684); it does not support IBM fixed-function retail controllers. The 4684 store system is supported only when the Retail Industry Program Support Services (RIPSS) version 2.0 or later software is being used in the 4684 controller. RIPSS is a program that provides access to the application files on the 4684 controller.

Retail Communications API

The RAIS base option consists of the tools needed for the communication to the store applications. At the heart of this is the RAIS Communications API. The QZCFPCMN service program allows for API access to retail application files over LU0 and TCP protocols. The API also uses the AS/400 Integrated File system support for retail application files that reside on the AS/400. The API can determine the protocol, or the user may specify which protocol should be used. The LU0 connection service communicates with the 4690 HCP. The TCP connection service communicate with 4690 Retail Connectivity Option. The Local connection service is a direct interface with the AS/400 Integrated File System.

Advanced Data Communication for Stores (ADCS) Emulation

ADCS emulation allows the AS/400 system to appear as an ADCS host system to a retail controller. RAIS imbeds the ADCS commands in the LU0 services of the communications API. The application suite enables the AS/400 user to emulate ADCS function and communicate with the HCP program in the retail controller:

- DUMP** Copies a **keyed file**¹ or **sequential file**² from the retail controller to an AS/400 database file. You can copy the entire file or part of it.
- LOAD** Loads a retail file from an AS/400 database file to a retail controller. You can load the entire file or part of it. The correct formatting of the file is the user's responsibility. The retail file conversion system can be used to help format the retail files.

¹ Type of file composed of keyed records. Each keyed record has two parts: a key and data. A key is used to identify and access each record in the file.

² A disk file in which records are read from or placed into the file according to the order they are processed. The term sequential file also includes direct and random access files.

ZERO	Clears (initializes) a keyed file on a retail controller. For 4690 files, ZERO clears the file data only. The performance statistics in the header sector are not reset. If the performance statistics must also be reset, you should load the file with an empty file by using LOAD instead.
CREA	Creates the space for a keyed or sequential file on a retail controller.
PURG	Deletes a file from a retail controller.
RED	Resets the end-of-data pointer in a sequential file on a retail controller to the beginning of the file. This erases all the data from the sequential file.
SUP	Starts a user program running, in background mode, on a retail controller.
KEYD	Sends a series of keyed record commands from the AS/400 system to a point-of-sale keyed file. The commands supported are ADD, REPLACE, DELETE, and READ record. The READ command is supported only through the ADCS emulation application programs interface (API).

Host Command Processor Emulation (HCP Emulation)

The **host command processor (HCP)** emulation is a function of the Retail Application Interface Suite that allows the AS/400 system to appear to an ADCS host system as if the AS/400 system were the host command processor in the point-of-sale system. It allows the AS/400 system to receive ADCS commands from the host system and respond as if it were the retail controller. HCP emulation supports the following ADCS commands from the host system:

DUMP	Sends sector images of a keyed or sequential file from an AS/400 database file to the host system. You can send the entire file or part of it.
LOAD	Loads sector images of a keyed or sequential file from the host system to an AS/400 database file. You can load the entire file or part of it.
ZERO	Clears (initializes) a simulated keyed file on the AS/400 system.
CREA	Creates the space for a simulated keyed or sequential file on the AS/400 system.
PURG	Deletes a file from the AS/400 system.
RED	Resets the end-of-data pointer in a sequential file on an AS/400 system to the beginning of the file. This erases all the data from the sequential file.
SUP	Starts a user program running on the AS/400 system.

The HCP emulation program has a host file cross-reference list that maps the point-of-sale file names sent by the host system to actual AS/400 file and library names.

Retail File Conversion

The RAIS file conversion is accomplished through retail conversion templates. A **template** is a description of the data mapping between a particular retail file and its equivalent AS/400 file or relational database table. The template is used to convert data from one format to the other. Examples for many of the IBM General Sales and Supermarket checkout applications files are provided in this guide. You can create additional templates or change those provided to satisfy your unique requirements.

The retail file conversion provides the option to generate COBOL/400*, RPG/400*, and data description specifications (DDS) source statements for inclusion in user application programs. These source statements describe the retail files after conversion to AS/400 database format. File conversion can be requested from an interactive menu, through AS/400 commands, or through the RAIS polling application.

The Retail Application Interface Suite also provides an optional archive scheme for point-of-sale data retrieved from stores. The point-of-sale data can be stored on the AS/400 system with store number and date as part of the file member name. The file conversion support has options to retrieve data by specific store and date from the main storage. It can also merge data from all stores into a single AS/400 database file. The flexibility provided by this archive scheme makes it much easier to manage and access data from a network of retail stores.

Retail File Polling Application

The polling application allows for **real time** polling of retail files. This function uses the retail communications API to access the retail files at the store location. It also uses the conversion templates to immediately convert the retail data and store it in an AS/400 flat file or relational database collection if desired. The user can define a list of files and store locations which this application will poll.

The polling application also provides an optional archive file access scheme for the transaction log data files.

Relational database collection and template support

RAIS includes an optional GSA relational database collection and template.

At install time of option 10, the system creates a relational database collection for the 4690 General Sales Application(GSA) and a corresponding template which maps the GSA transaction log files. The polling application uses the template to convert the transaction log into this relational database. The relational database then can be queried by an application to extract the important retail data.

Retail Application Logical Interface definitions for the GSA relational database

The Retail Application Logical Interface is a logical interface to the optionally installed GSA relational database collection. This support allows for consistent and easy extraction of the important retail transaction log data. The user can write applications to this interface or use interactive SQL to extract the data. Also included are a set of reports that use the Retail Application Logical Interface.

Point-of-Sale Environments

The AS/400 system operates in the point-of-sale environments as:

- In-store processor
- Host system

AS/400 System as an In-Store Processor

In the in-store processor environment, the AS/400 system serves as the connecting node between the retail controller and the host system. The AS/400 system communicates with the retail controller using the retail communications support.

If the host system is running the ADCS licensed program, the Retail Application Interface Suite HCP emulation can be used to emulate a retail controller. In this environment, the AS/400 **SNA upline facility (SNUF)**, the communications support that allows the AS/400 system to communicate with CICS/VS and IMS/VS application programs on a host system, is used to communicate between the host system and the AS/400 system. The HCP emulation function of the Retail Application Interface Suite provides the data transmission functions with the host ADCS and allows the in-store processor to be inserted into an existing ADCS point-of-sale network without rewriting any of the host communications programs.

If the host system is not using ADCS, then advanced program-to-program communications (APPC), NetView* Dis-

tribution Manager (NetView DM), Distributed Distribution Manager, AS/400 object distribution, or other AS/400 communications methods can be used to transfer files between the in-store AS/400 system and the host system.

Retail Application Interface Suite can:

- Provide the transmission of data between the AS/400 in-store processor and the IBM store system through the ADCS emulation function.

- Provide the transmission of data between the AS/400 in-store processor and the ADCS host system using the HCP emulation function.

- Allow user applications to process point-of-sale data files while they reside on the AS/400 system.

- Convert retail file data into AS/400 file data and AS/400 data files into retail data files using the file conversion function.

The AS/400 retail communications support pass-through function can also be used between the retail controller and the host processor in this environment to provide:

- Electronic Fund Transfer (EFT)

- ADCS (pass-through)

- NetView DM (pass-through)

- User application program-to-program communications

AS/400 applications do not have access to data flowing through the AS/400 in-store processor when using the retail pass-through support.

The AS/400 system provides the power and flexibility to run many additional user applications in the store environment. Inventory management, time and attendance, and pharmacy are some additional applications that could be considered.

AS/400 Retail Systems In-Store Processor Environment: Figure 1-1 on page 1-4 illustrates the parts of AS/400 retail communications showing the AS/400 system as an in-store processor. Following the figure are descriptions of the parts.

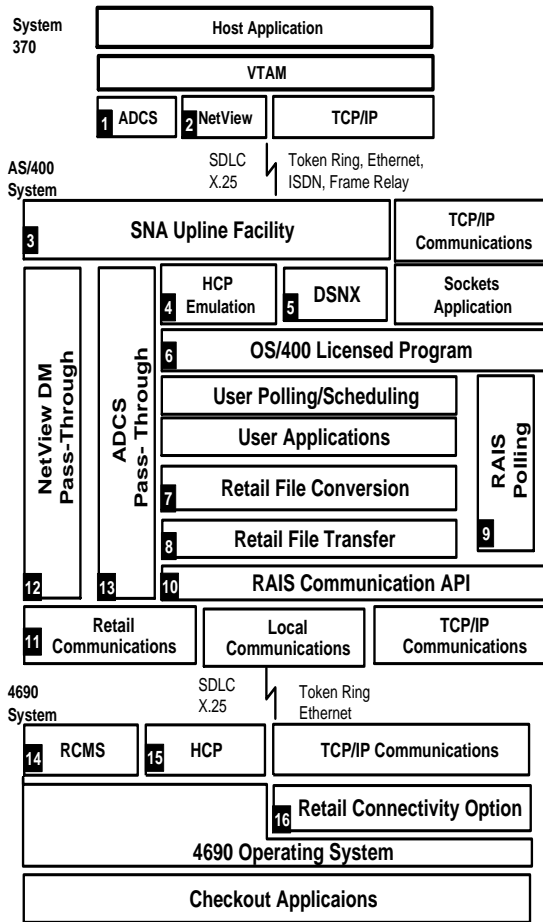


Figure 1-1. AS/400 Retail Systems In-Store Processor Environment

- 1 ADCS: **Advanced Data Communications for Stores (ADCS)** is an IBM host polling licensed program written for retail networks. It sends HCP commands to the host command processor of the retail controller to obtain the desired data transmission. ADCS is a licensed program that functions as the host processor for host system to point-of-sale system communications. The point-of-sale system is a retail environment system consisting of a store controller and one or more retail devices.
- 2 NetView DM: **NetView Distribution Manager** is a licensed program that gives you the capability to plan, schedule, and control the exchange of data between the host system and one or more remote sites. Distributed systems node executive (DSNX) support allows the AS/400 system to be part of the network.
- 3 SNA upline facility: SNUF is an AS/400 communications system function that provides SNA logical unit type "0" secondary node support.
- 4 HCP emulation: As a function of the Retail Application Interface Suite, HCP emulation allows the AS/400 system to appear as a host command processor to the host system. This allows the AS/400 system to run ADCS commands sent from the host system.

5 **Distributed systems node executive (DSNX):** DSNX is a function of the operating system that receives and analyzes requests from the NetView Distribution Manager licensed program on a host system. If the request is directed to the system that receives it, the request is processed on that system or on a personal computer directly attached to that system. If the request is intended for a different system, it is routed toward its destination.

6 AS/400 licensed program: Version 4, Release 3 or later of the IBM Operating System/400 licensed program.

7 Retail file conversion: Provides easy-to-use conversion for retail controller data files.

8 Retail file transfer: Retail file transfer commands allow the AS/400 system to appear as a host system to the host command processor, or Retail Connectivity Option program on a retail controller. The commands allow either an AS/400 user, through an AS/400 display station, or a user-written application program to send file transfer commands to a retail controller. The Retail file transfer commands are written to the RAIS API.

9 Retail Polling Application: Provides real time data transfer and conversion for point-of-sale data files. The template support allows for the population of a flat file, user created relational database collection or the pre-defined GSA relational database.

1 RAIS Application Programming Interface: The RAIS API is a retail communications API that communicates with the Host Command Processor, Retail Connectivity Option on the retail controller, or the Integrated File System on the AS/400. This allows an application to be written independent of the protocol which will be used.

11 AS/400 communications support: The operating system support provides synchronous data link control (SDLC) SNA, X.25/SNA, Token Ring, Ethernet, or Wireless communications with the IBM retail controller. The protocols used may be logical unit "0" or TCP/IP. These features are part of the AS/400 system and enables the communications programmer to write point-of-sale communications applications without using the Retail Application Interface Suite. For logical unit "0," this level of programming requires knowledge of HCP and inter-system communications function (ICF). For TCP/IP, this level of programming requires knowledge of sockets.

12 NetView DM pass-through: As a function of the AS/400 retail communications support, NetView DM pass-through allows a System/370* host system running NetView DM to communicate directly to the remote change management server in the IBM retail controller through an intervening AS/400 in-store processor. The data passing through the AS/400 system is not available to AS/400 applications.

- 13 **ADCS pass-through:** As a function of the AS/400 retail communications support, ADCS pass-through allows an ADCS host system to communicate directly to an IBM retail controllers HCP, through an intervening AS/400 in-store processor. The data passing through the AS/400 system is not available to AS/400 applications.
- 14 **RCMS: Remote change management server (RCMS)** is a function of the 4690 Store System, running commands sent by NetView DM on a System/370 host system. It is a store controller program that communicates over a network that connects a host processor and a store controller. RCMS allows the NetView Distribution Manager to access point-of-sale controller files. It also provides error reporting and recovery for failures and data format conversion for files.
- 15 **HCP:** As part of the IBM store operating system, the HCP program resides in the retail controller and runs ADCS commands from a host system or in-store processor. HCP provides a host controlled file management system for retail controller files.
- 16 **Retail Connectivity Option:** An optional program product which provides TCP/IP connectivity service for hosts to the retail store file system.

AS/400 System as Host System for a Point-of-Sale Network

In this environment, the AS/400 system serves as the host system for the point-of-sale network. Multiple store sites can be controlled from a single AS/400 host site. The point-of-sale network is usually a switched telephone network, but non-switched telephone lines are also supported. For 4684 controllers, the network can also be through a X.25 permanent virtual circuit (PVC) or a switched virtual circuit (SVC).

In a switched-line environment, the AS/400 system can use automatic call to dial each store in the network for uploading sales information, downloading price changes, or for whatever data transfer is desired.

In a non-switched or X.25 line environment, the retail user can use a single line to the store for all point-of-sale file transfer communications, and for host credit, 5250, and 3270 emulation. The 4690 Store System supports concurrent 3270 emulation and file transfer communications on the same 4690 controller. This 3270 emulation would allow AS/400 applications to access ASCII work stations attached to the 4690 store controller.

The AS/400 system communicates with the IBM store systems using retail communications support. The ADCS emulation function of the Retail Application Interface Suite provides the data transfer function for transfers to the IBM store system. The AS/400 system provides both two-line automatic call (Bell 801 Type) and single-line automatic dial (V.25bis) to facilitate polling in a switched network. Addi-

tional user programming is required to provide the scheduling and error recovery necessary for a complete host polling package.

The Retail Application Interface Suite also provides an optional archive scheme for data. The file conversion system has options to retrieve data from the archive by specific store and date.

AS/400 Retail Systems Host Environment:

Figure 1-2 illustrates the components of AS/400 retail communications showing the AS/400 system as a host system. Following the figure are descriptions of the components.

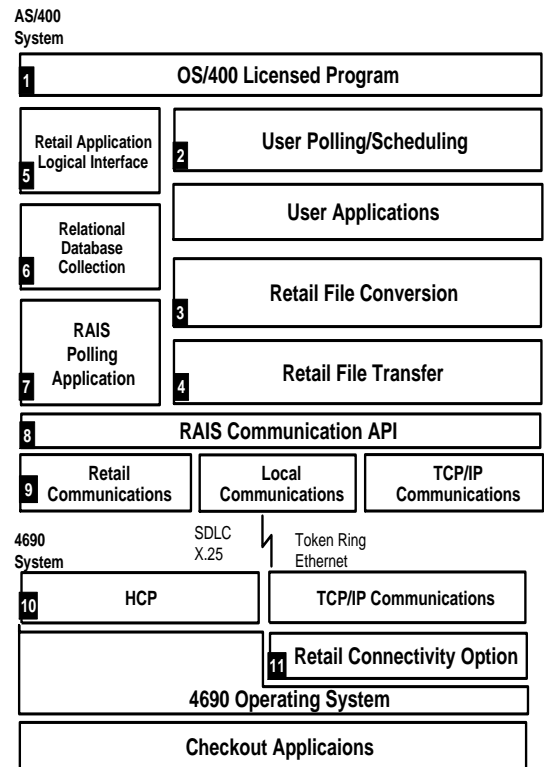


Figure 1-2. AS/400 Retail Systems Host Environment

- 1 **AS/400 licensed program:** Release 4.3 or later of the IBM Operating System/400 licensed program.
- 2 **User polling/scheduling:** The AS/400 system provides both two-line automatic call (Bell 801 Type) and single-line automatic dial (V.25bis) to facilitate polling in a switched network. Additional user programming is required to provide the scheduling and error recovery necessary for a complete host polling package.
- 3 **Retail file conversion:** Provides data conversion for point-of-sale data files.

- 4 Retail file transfer: Retail Application Interface Suite for AS/400, Retail file transfer commands allows the AS/400 system to appear as a host system to the host command processor, or Retail Connectivity Option program on a retail controller. The commands allow either an AS/400 user, through an AS/400 display station, or a user-written application program to send file transfer commands to a retail controller. The Retail file transfer commands are written to the RAIS API.
- 5 Retail Application Logical Interface: The Retail Application Logical Interface is created over the GSA relation database collection. The logical interface is used to extract data from the GSA relational database collection excluding unwanted data (voids and training transactions).
- 6 Relational Database collection: A predefined relational database collection for the GSA TLOG which can be populated using the retail polling application.
- 7 Retail Polling Application: Provides real time data transfer and conversion for point-of-sale data files. The template support allows for the population of a flat file, user created relational database collection or the predefined GSA relational database.
- 8 RAIS Application Programming Interface: Retail Application Interface Suite for AS/400, The RAIS API is a retail communications API that communicates with the Host Command Processor, Retail Connectivity Option on the retail controller or the Integrated File System on the AS/400. This allows an application to be written independent of the protocol which will be used.
- 9 AS/400 communications support: The operating system support provides synchronous data link control (SDLC) SNA, X.25/SNA, Token Ring, Ethernet, or Wireless communications with the IBM retail controller. The protocols used may be logical unit "0" or TCP/IP. These features are part of the AS/400 system and enables the communications programmer to write point-of-sale communications applications without using the Retail Application Interface Suite. For logical unit "0," this level of programming requires knowledge of HCP and intersystem communications function (ICF). For TCP/IP, this level of programming requires knowledge of sockets.
- 1 HCP: As part of the IBM store operating system, the HCP program resides in the retail controller and runs ADCS commands from a host system or in-store processor. HCP provides a host controlled file management system for retail controller files.
- 1.1 Retail Connectivity Option: An optional program product which provides TCP/IP connectivity service for hosts to the retail store file system.

Planning for Retail Communications

Planning your configuration consists of considering all parts of your network and the types of interactions to occur between those parts. For example, the values entered when configuring for the Retail Application Interface Suite depend on the values entered when configuring the retail controllers.

Configuring the communications utility also depends on whether the AS/400 system is a host system or an in-store processor. If the AS/400 system is the host system, you need to consider the configuration of retail communications only. If the AS/400 system is an in-store processor, or you are using retail pass-through, then retail communications and SNA upline facility (SNUF) configurations must be considered. For TCP/IP protocol communications, you should additionally consider the Retail Connectivity Option server installation and configuration on the 4680 and 4690 controllers.

For further information on communication configurations, see the following resources.

Communications Configuration, SC41-5401-00 book

TCP/IP Configuration and Reference, SC41-5420-01 book

the appropriate retail controller book for the controller to be configured.

Planning and configuration should be performed by someone who is familiar with the following:

AS/400 system operations

AS/400 system communications

The planned point-of-sale operating environment

Retail controller configuration

Planning Requirements

The planner should ensure the AS/400 system has been properly installed and that all of the minimum hardware and software requirements are present for the planned operating environment.

Hardware Requirements: If the AS/400 system is going to be operating as an in-store processor, the minimum hardware requirements are the following:

AS/400 system with at least 8MB (MB equals 1 048 576 bytes) main storage

Communications cables

Communications line to a host system can be synchronous data link control (SDLC) (switched or nonswitched), X.25, token-ring, ethernet, frame relay, distributed data interface(DDI). or integrated services digital network (ISDN).

One ISDN data link control (IDLC), SDLC, X.25, token-ring, ethernet, frame relay, or DDI communications line for host connection

One SDLC or X.25 communications line for connecting the Point-of-Sale controller, a real time interface

coprocessor multiport adapter, a token-ring adapter card, an ethernet adapter card, or a wireless adapter card.
Modems or modem eliminators for communications lines

If the AS/400 system is going to be operating as a host, the minimum hardware requirements are the following:

AS/400 system with at least 8MB main storage
Communications cables
One or more SDLC, X.25, token-ring or ethernet communications lines for connecting the point-of-sale controllers.
Modems or modem eliminators for communications lines

Note: The retail controller will need an SDLC card or multiple processor communications adapter (MPCA) card for communication with the AS/400 system. For X.25, the retail controller requires an X.25 card. For token-ring connections, a token-ring adapter card is required. For ethernet connections, an ethernet adapter card is required. For wireless connections, a wireless adapter card is required.

Software Requirements: If the AS/400 system is going to be operating as an in-store processor with the System/370* host system running ADCS, the software requirements are the following:

For the AS/400 system:

IBM Operating System/400 licensed program 5769-SS1, Version 4 Release 3 Modification 0

To use the Relational Database options of RAIS, order the DB2 UDB for AS/400 Group PTF for your &ss1s0. release. This can be ordered via Electronic Customer Support (ECS) or by contacting IBM service.

For the System/370 host system:

Advanced Data Communication for Stores (ADCS) licensed program 5735-XR2, Release 5, Modification Level 2

The IBM 4690 Operating System

- Version 1, license program number 5696-538, with the TCP/IP the IBM 4690 TCP/IP Feature (5764-064)
- The IBM 4690 Operating System, Version 2, license program number 5696-F53

To use the TCP/IP interface, the IBM 4690 Retail Connectivity Option (5799-QXG)

For 4684 store controllers:

Retail Industry Program Support Services (RIPSS), Release 2 or later. RIPSS provides the HCP communications software support for the 4684 store controller. (Release 3.01 is needed if running X.25.)

For 3650 programmable store systems retail controllers:
Component ID 5668943; PTF level 349863

For 3680 programmable store systems retail controllers:
Component ID 5668942; PTF levels 320503, 320560, and 320561

If the AS/400 system is going to be operating as a host system, the minimum software requirements are:

For the AS/400 system:

IBM Operating System/400 licensed program 5769-SS1, Version 4 Release 3 Modification Level 0

To use the Relational Database options of RAIS, order the DB2 UDB for AS/400 Group PTF for your &ss1s0. release. This can be ordered via Electronic Customer Support (ECS) or by contacting IBM service.

The IBM 4690 Operating System

- Version 1, license program number 5696-538, with the TCP/IP the IBM 4690 TCP/IP Feature (5764-064)
- The IBM 4690 Operating System, Version 2, license program number 5696-F53

To use the TCP/IP interface, the IBM 4690 Retail Connectivity Option (5799-QXG)

For 4684 store controllers:

Retail Industry Program Support Services (RIPSS), Release 2 or later. RIPSS provides the HCP communications software support for the 4684 store controller. (Release 3.01 is needed if running X.25.)

For 3650 programmable store systems retail controllers:

Component ID 5668943; PTF level 349863

For 3680 programmable store systems retail controllers:

Component ID 5668942; PTF levels 320503, 320560, and 320561

Installation Information

The Retail Application Interface Suite is comprised of several application each being an option of the 5X99RAI suite. The options are separated into 2 features. The 5001 Feature includes the Base option(00) and Point of Sale/2000 option(01). The 5002 Feature includes the Polling Application option(02), the General Sales Application Database option(10) including the Retail Application Logical Interface for GSA, the Supermarket Application Database option(11) including the Retail Application Logical Interface for SA, and the RALI application option(50).

When RAIS is ordered you will receive both features that can be installed 1 of 2 ways. The **LODRUN** can be used on tape or CD media to install each of the features options. Otherwise each option can be installed separately.

Installing the Utility - LODRUN

The installation of the Retail Application Interface Suite can be accomplished by completing the following steps:

1. Sign on to your AS/400 system as the security officer (QSECOFR).
2. Install the utility using the LODRUN command. Specify the following options:

```
LODRUN DEV(device-name)
```

This will install each of the Features options:

- *BASE - RAIS base option
- 1 - RAIS Point of Sale/2000 option
- 2 - RAIS Polling Application option
- 10 - General Sales Application relational database and template definition including the Retail Application Logical Interface(RALI) for GSA
- 11 - Supermarket Application relational database and template definition including the Retail Application Logical Interface(RALI) for SA
- 50 - RALI Applications and Reports

Note: Depending on what you ordered, either the GSA Database option(10) or the SA Database option(11) will be packaged with the 5002 feature.

Installing the Utility options

The installation of the Retail Application Interface Suite can be accomplished by completing the following steps:

1. Sign on to your AS/400 system as the security officer (QSECOFR).
2. Install the utility using the RSTLICPGM command. Specify the following options:

```
RSTLICPGM LICPGM(5X99RAI) DEV(device-name)  
OPTION(optional-part)
```

where optional-part = :

- *BASE - RAIS base option
- 1 - RAIS Point of Sale/2000 option
- 2 - RAIS Polling Application option
- 10 - General Sales Application relational database and template definition including the Retail Application Logical Interface(RALI) for GSA
- 11 - Supermarket Application relational database and template definition including the Retail Application Logical Interface(RALI) for SA
- 50 - RALI Applications and Reports

Note: The restore will attempt to restore the Primary language version for this product. If your system

does not have a supported primary language, no language will be restored. Use the following to install the 2924 language feature:

```
RSTLICPGM LICPGM(5X99RAI) DEV(device-name)  
OPTION(optional-part)  
LNG(2924)
```

This will install the 2924 language option into the QRAIS2924 library. This library MUST be placed in your library list.

3. Sign off your AS/400 system as the security officer (QSECOFR).

After you have completed the installation, the following will be accomplished:

A new library named QRAIS will now be on your system.

All code objects will be restored to this library.

Conversion templates, store information, and host cross reference information configuration files are duplicated in QUSRSYS. The following files are duplicated in library QUSRSYS:

- Conversion template physical files (QAZCFXHDR, QAZCFXRCD, and QAZCFXFLD)
- Store information physical file (QAZCFSTOR)
- Store polling entry physical file (QAZCFTPOL)
- Retail file table entry physical file (QAZCFRFTTE)
- Host cross reference physical file (QAZCFXREF)

If option 10 or 11 is installed, a new library named QUSRPOSyy will now be on your system and a new template definition named QZCFyyDB will be installed into the template configuration files.

where yy = :

GS - General Sales Application relational database collection

SA - Supermarket Application relational database collection

You do not need to add the QUSRPOSyy library to your library list when using the retail polling application. You must however add the library to your library list when using the RMVRTLTNS or the PRTRTLRPT commands. See Chapter 5, "RAIS Retail Polling Application" on page 5-1 for additional polling application information.

Note: It is possible to have the 5769CF1 product installed with 5X99RAI. In this case, unless you library-specify all the commands, you should place the QRAIS library, or the Language Library(if it was not installed into QRAIS) in front of the QSYS library in the system library list.

A new user profile (QRAIS) and authorization list (QRAIS) will be on your system. These are used with HCP emulation and the polling application.

Adding the RAIS License Key

RAIS is created as a licensed offering. In order to use the Communications API, POSA/2000 commands, or the Polling Application you must first add either a system license key or enterprise license key.

A system license key is generated by your RAIS software provider using the system serial number where the offering resides. Once you receive the license key from your provider, the ADDLICKEY command should be used to add the license for this system.

```
ADDLICKEY PRDID(5X99RAI) LICTRM(license_term)
          FEATURE(feature)
          SERIAL( LOCAL)
          LICKEY(XXXXXX YYYYYY ZZZZZZ)
          USGLMT( NOMAX) EXPDATE(expiration_date)
```

Your RAIS software provider may provide you with a License Key file. A license key file is used for a group of systems. You will have to provide the serial numbers for each system in this group. The license key for each will be generated and stored in this file. The ADDLICKEY command should be used to add licenses for this system.

```
ADDLICKEY PRDID(5X99RAI) LICTRM(license_term)
          FEATURE(feature)
          SERIAL( LOCAL)
          LICKEYFILE(library/file_name)
          LICKEYMBR(member_name)
          USGLMT( NOMAX) EXPDATE(expiration_date)
```

An enterprise license key is generated by your RAIS software provider using a Master serial number for your systems. This could be any system serial number in your enterprise. Once you receive the license key from your software provider, the QRAIS/ADDENTLCKY command should be used to add the license on all systems in your enterprise.

```
ADDENTLCKY PRDID(5X99RAI) LICTRM(license_term)
           FEATURE(feature)
           SERIAL(enterprise_serial_number)
           LICKEY(XXXXXX YYYYYY ZZZZZZ)
           EXPDATE(expiration_date)
```

Where

license_term The 5X99RAI version license term. This is supplied by your provider.

feature The 5X99RAI feature that is licensed. The 5001 feature is comprised of the Base option and option 1. The 5002 feature is comprised of options 2, 10, 11, and 50. Each feature is License Key enabled.

XXXXXX YYYYYY ZZZZZZ The generated license key for your system or master serial number. This is supplied by your provider.

library/file_name The file name where the license key member resides. This is only used when a group of systems is enabled through a license key.

member_name The member name in the license key file that has the generated license key for your local system.

enterprise_serial_number The master serial number for your enterprise.

expiration_date The expiration date of the license key. This should be set to *NONE unless otherwise noted.

Saving the Utility

The Retail Application Interface Suite is saved by completing the following steps:

1. Sign on to your AS/400 system as the security officer (QSECOFR).
2. Initialize the tape that will hold the utility using the INZTAP command.
3. Save the utility using the SAVLICPGM command. Specify the following options:

```
SAVLICPGM LICPGM(5X99RAI) DEV(device-name)
          OPTION(optional-part)
```

where optional-part = :

*BASE - RAIS base option

1 - RAIS Point of Sale/2000 option

2 - RAIS Polling Application option

10 - General Sales Application relational database and template definition including the Retail Application Logical Interface(RALI) for GSA

11 - Supermarket Application relational database and template definition including the Retail Application Logical Interface(RALI) for SA

50 - RALI Applications and Reports

Where *device-name* specifies the tape device containing the tape which the licensed program will be saved.

4. Sign off your AS/400 system as the security officer (QSECOFR).

Note: Conversion templates, store information, and host cross reference information are not saved when the product is deleted. If any of this information has been added or changed and should be saved, the following files need to be saved in library QUSRSYS:

Conversion template physical files (QAZCFXHDR, QAZCFXRCD, and QAZCFXFLD)
 Store information physical file (QAZCFSTOR)
 Store polling entry physical file (QAZCFTPOL)
 Retail file table entry physical file (QAZCFRFTE)
 Host cross reference physical file (QAZCFXREF)

Also, if the GSA Relational Database Collection has been installed by option 10, the data will not be saved. You must save off the data libraries: QUSRPOSGS for the Option 10 installation.

Deleting the Utility

The Retail Application Interface Suite is deleted by completing the following steps:

1. Sign on to your AS/400 system as the security officer (QSECOFR).
2. Remove QRAIS and any Language Library(if necessary) from all library lists.
3. Delete the utility using the DLTLICPGM command. Specify the following options:

```
DLTLICPGM LICPGM(5X99RAI)
OPTION( ALL)
```
4. Check for failures during the deletion using the DSPJOBLOG command to see if any Retail Application Interface Suite objects were not deleted from the system.
5. Sign off your AS/400 system as the security officer (QSECOFR).
6. The Relational Database collections installed with Option 10 will not be deleted. The user must delete these Collections if they are no longer needed. The collection name is QUSRPOSGS for Option 10.

Deleting the Retail Application Interface Suite will cause all items placed in the AS/400 system during the install process to be deleted. The following configuration files will not be deleted from QUSRSYS.

Conversion template physical files (QAZCFXHDR, QAZCFXRCD, and QAZCFXFLD)
Store information physical file (QAZCFSTOR)
Store polling entry physical file (QAZCFTPOL)
Retail file table entry physical file (QAZCFRFTE)
Host cross reference physical file (QAZCFXREF)

If the point-of-sale utility is re-installed, the files will not be affected. If the files are to be re-installed, they must first be deleted from library QUSRSYS using the DLTF command. The install procedure will not overlay existing files in library QUSRSYS.

Retail Application Interface Suite effect on customer applications

Retail Application Interface Suite is designed to use the same commands as the V4R2M0 version of the Point-of-Sale Communications Utility(5769CF1 Licensed Program Product). It also uses some of the same Menu objects. This allows Current 5769CF1 user designed applications to use the

Retail Application Interface Suite without having to change code or compile.

Depending on how an application was designed, there may be some user intervention required.

Retail Application Interface Suite integration with customer applications

QRAIS or the Language Library(If it was not installed into QRAIS) should be placed in the library list.

This applies to applications that were designed to use the Point-of-Sale Utility commands from the library list.

QPOS library must be removed from applications.

This applies to applications that were designed with QPOS or QSYS library specified commands. These must be replaced with the QRAIS or the Language Library(If it was not installed into QRAIS) library specification.

Applications may not use the QART* template files.

Retail Application Interface Suite template support uses the QAZCF* files in QUSRSYS only. There is no library list support for these files. Applications that are expecting the RAIS conversion commands to use templates not in the QUSRSYS library will not work.

Conversion processing for templates defined using direct(4) Retail file organization and Variable length fields has changed.

Specifying (Y)es for the Variable length fields on Templates defined as Direct file organization in The Point-of-Sale Utility and no affect on the conversion processing. This has changed in Retail Application Interface Suite. If a Direct file template is defined with Variable length fields the Convert To Retail File command will check the Calculated record length of the Retail file and compare it to the Retail record length defined in the template. If the Retail record length is at least 2 Greater than the calculated length, a CRLF(0D0A) delimiter will be set at the end of the Record. For the old support, the Variable length fields parameter must be set to no.

Retail Application Interface Suite location manager concerns

User applications that use AS/400 commands that require a remote location or device or controller description parameters will need to be changed.

Retail Application Interface Suite uses a location manager to determine protocol and find the configuration object to use. These configuration objects may not be remote location names, or Device Configuration objects. For the TCP protocol, the Retail Application Interface Suite location is a Host Configuration name. The location can also be specified as a Retail Store Information entry store number. When this is used, the store entry locations are used to determine the location.

Applications using the QRTLAPI API program will only be able to access LU0 locations for requests other than acquire.

The QRTLAPI has not changed for the ADD, DEL and READ functions. For these functions the LU0 locations are the only ones supported. Any applications using these functions, that need to access TCP locations, must be changed to use the Retail Application Interface Suite API. See Chapter 3, "Retail File Application Programming Interfaces" on page 3-1 for more information on this topic.

Retail Application Interface Suite coexistence with the Point-of-Sale Utility (5769CF1)

Library QRAIS or the Language Library (If it was not installed into QRAIS) should be placed in front of QSYS in the library list.

The 5769CF1 LPP installs the commands and menus in the QSYS library. If a user or application is to use the Retail Application Interface Suite QRAIS or the Lan-

guage Library (If it was not installed into QRAIS) commands must be before the QSYS commands, or the commands must be library specified.

QPOS library must be removed from applications.

If an application was designed with QPOS or QSYS library specified commands, these must be replaced with the QRAIS or the Language Library (If it was not installed into QRAIS) library specification. If QRAIS or the Language Library (If it was not installed into QRAIS) is placed in front of QSYS, then the library specification is not needed.

HCP emulation will use the **qsys/qrthcpem** program

If the 5769CF1 LPP does exist on the system HCP emulation will continue to use the **qsys/qrthcpem** program for processing. If you plan on deleting the 5769CF1 LPP and use the HCP emulation of RAIS, please see "Using Host Command Processor (HCP) Emulation" on page 4-24 for additional configuration information.

Chapter 2. Getting Started with the Retail Application Interface Suite Base Option

This chapter contains information for the Retail Application Interface Suite base option for:

- Getting started with the utility
- Configuring for retail communications
- Configuring for retail conversion
- Maintaining retail files and lists

Getting Started with the Retail Application Interface Suite

Use the Retail Application Interface Suite for AS/400 (RAISCTL) menu to get started. Type:

```
GO RAISCTL
```

and press the Enter key to display this menu.

```
RAISCTL      Retail Application Interface Suite for AS/4
                                     System:  RCHAS4
Select one of the following:
  1. Work with retail communications
  2. Work with retail file conversion
  3. Configure retail communications application
```

Option 1 (Work with retail communications):

This option accesses the Work with Retail Communications menu which allows you to use the various Retail Application Interface Suite communications functions. Displays and information relating to this option are located in "Working with Retail Communications" on page 2-4.

Option 2 (Work with retail file conversion):

This option activates the Work with Retail File Conversion menu which allows you to work with the various conversion functions. Displays and information relating to this option are located in "Retail File Conversion" on page 2-7.

Option 3 (Configure retail communications application):

This option accesses the Configure Retail Communications Application menu which allows you to perform configuration and maintenance functions. Displays and information relating to this option are located in the next section.

Configuring for Retail Communications

This section contains the configuration and maintenance displays that are accessed through the Retail Application Interface Suite to prepare for retail communications. Information on configuring the AS/400 system for retail communications can be found in the *Retail Communications Programming* book.

Configure Retail Communications Application Menu

Selecting option 3 (Configure retail communications application) from the RAISCTL menu will display this menu.

This menu provides access to the configuration options and to the AS/400 system's communications configuration menu.

```
RAISCFG      Configure Retail Communications Application
                                     System:  RCHAS4
Select one of the following:
  1. Configure communications
  2. Work with retail store information
  3. Work with host file cross reference
  4. Work with retail polling entry
  5. Work with retail file table
```

Option 1 (Configure communications):

Use this option to access the standard AS/400 communications configuration menu. You must configure the lines, controllers and devices for the SNA protocol, and an interface and routes for the TCP protocol. (see configuration examples in "Retail Controller Configuration Examples").

Option 2 (Work with retail store information):

Use this option to provide maintenance for the store information file.

Option 3 (Work with host file cross reference):

Use this option to maintain the host cross-reference file list used by HCP emulation on the AS/400 system. The AS/400 host file cross-reference maps the 6-character ADCS file name into the AS/400 file and library combinations specified.

The host cross-reference file list is only used by the HCP emulation program while communicating to an ADCS host system.

Option 4 (Work with retail polling entry):

Use this option to provide maintenance for the store polling entries, and to see the status of the polling application entries.

Option 5 (Work with retail file table):

Use this option to provide maintenance for the retail file table and its entries.

Work with Retail Store Information

Selecting option 2 (Work with retail store information) from the Configure Retail Communications Application menu shows the Work with Retail Store Information display.

You can also use the WRKRTLSTOR command to access this display.

This display provides maintenance functions for the retail store information file. This file contains the store number and related information for each retail location.

```

Work with Retail Store Information

Type options, press Enter.
 2=Change  4=Delete  5=Display  7=Rename

      Store
Option  Number  Store Description  Store Telephone Number
-      2323  Store 2323 New York, NY.  (3 1) 555-1234

F3=Exit      F5=Refresh      F6=Add store information
F11=Display managers/locations  F12=Cancel  F21=Print list
    
```

The retail store information file is used to provide additional information about each retail store controller. This information is useful when a network of retail controllers is managed from a single AS/400 location.

This is the source of the store number used in the *STORE option of the Receive Retail File (RCVRTLF) command and the store information shown on the Work with Retail Communications display.

The store number is used in the Retail Application Interface Suite archive scheme. This number is combined with the job date to form the member name of received retail data. See the Receive Retail File command *STORE option for details of the member name format when using the *STORE option.

By pressing F11 (Display managers/locations) on the Work with Retail Store Information display, the information on the display is changed to provide manager and remote locations information:

```

Work with Retail Store Information

Type options, press Enter.
 2=Change  4=Delete  5=Display  7=Rename

      Store
Option  Number  Store Manager      Primary Location
-      2323  George A. Smith    HCPC1
-      13    George A. Smith    RWCCHCP
-      3    George A. Smith    PRIMARY469 .TESTNADE.I..

F3=Exit      F5=Refresh      F6=Add store information
F11=Display descriptions  F12=Cancel  F21=Print list
    
```

F21 (Print list) provides a printed list of the various retail store controllers and their associated remote location names.

Add Retail Store Information

Select F6 (Add store information) from the Work with Retail Store Information menu to access the Add Retail Store Entry Command.

This command is used to add a new retail store information record. The store number must be unique and 6 digits long. Leading zeros are permitted. Remote location names are validated through the Retail Application Interface Suite. Therefore, the Configuration objects must be created before the store information record is created. The Configuration name element and Alternate Configuration name element can be One of the following:

- Retail Device remote location name (LU0 support)
- TCP/IP Host name

The SUP Location / TCP port elements are used when the configuration name elements are for LU0 or TCP/IP support. For LU0, a Start User Program Retail device remote location can be specified for the 4690 SUP location. For TCP/IP, a TCP port can be specified for the Retail Connectivity Option server port.

Note: The TCP port will default to 20000 if it is not specified.

The Retail Store Information entries are used by the RAIS API for dual controller support. If the location used by the API is a store number found in this table, the Configuration name is used as the primary location and the Alternate Configuration name is used as the alternate location.


```

Add Retail Store Entry (ADDRTLSTOR)

Type changes, press Enter.

Store number . . . . . 2323 1-999999
Communication Protocol . . . . . IP SNA, IP, LOCAL
Primary Location . . . . . PRIMARYCTL.TESTNODE.IBM.COM

Alternate Location . . . . . ALTERNATECTL.TESTNODE.IBM.COM

Retail Store Manager . . . . . George A Smith
Retail Store Telephone Number . . . . . 321 555 1234
Retail Store Description . . . . . George's Hardware Store
SUP Location . . . . . Character value
TCP Port . . . . . 2 1-65535
Alternate SUP Location . . . . . Character value

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

```

Confirm Delete of Retail Store Information

Press Enter to confirm your choices for 4=Delete.
Press F12 to return to change your choices.

Store
Option Number Store Description Store Telephone Number
4 2323 Store 2323 New York, NY. (321) 555-1234

F11=Display managers/locations F12=Cancel

```

Change Retail Store Information

Select option 2 (Change) from the Work with Retail Store Information menu to access this display. This display is used to change the retail store information.

```

Change Retail Store Information

Store number . . . . . : 2323

Type choices, press Enter.
Store description . . . . . George's Hardware Store
Store manager . . . . . George A Smith
Store telephone number . . . . . 321 555 1234
Store communication protocol . . . . . IP SNA, IP, LOCAL
Primary location . . . . . PRIMARYCTL.TESTNODE.IBM.COM

TCP port . . . . . 2 Number
Alternate location . . . . . ALTERNATECTL.TESTNODE.IBM.COM

Alternate TCP port . . . . . 2 1 Number

F3=Exit F12=Cancel

```

The Display Managers and Locations display provides managers and remote locations with information. It is selected by pressing F11 (Display managers/locations) on the Confirm Delete of Retail Store Information display.

```

Confirm Delete of Retail Store Information

Press Enter to confirm your choices for 4=Delete.
Press F12 to return to change your choices.

Store
Option Number Store Manager Primary Location
4 2323 George A. Smith HCPC1

F11=Display descriptions F12=Cancel

```

Confirm Delete of Retail Store Information

Select option 4 (Delete) from the Work with Retail Store Information menu to access this display.

This display is used to confirm the deletion of existing store information records.

Pressing the Enter key on either of the delete confirmation displays deletes all the shown store information records.

Display Retail Store Information

Select option 5 (Display) from the Work with Retail Store Information menu to access this display.

```

Display Retail Store Information
Store number . . . . . : 2323
Store description . . . . . : George's Hardware Store
Store manager . . . . . : George A Smith
Store telephone number . . . . . : 321 555 1234
Store communication protocol . . . . . : SNA
Primary location . . . . . : HCPCL

SUP location . . . . . : SUPCL
Alternate location . . . . . : HCPDL

Alternate SUP location . . . . . : SUPDL

Press Enter to continue.
F3=Exit F12=Cancel

```

Rename Retail Store Information

Select option 7 (Rename) from the Work with Retail Store Information menu to access this display.

This display is used to change the store number associated with an existing retail controller. Data stored previously with the *STORE archive option will still be under the previous store number, NOT the new store number.

```

Rename Retail Store Information
To rename store number, type New Number, press Enter.

Store Number   New Number
2323           _____

```

Retail Store Information Configuration commands:
The add retail store entry command is used to add store entries to the store information configuration table.

```

Job: B,I Pgm: B,I REXX: B,I Exec
Threadsaf: No

ADDRTLSTOR
STORENBR( retail-store-number )
PROTOCOL( SNA (P)
          IP
          LOCAL
PRILOC( primary-location )
ALTLOC( alternate-location )
STOREMGR( retail-store-manager )
STORETELE( retail-store-telephone-number )
STOREDESC( retail-store-description )
SUPLOC( start-user-program-location )
          2
PRIPORT( primary-tcp-server-port )
SUPALT( alternate-sup-location )
          2
ALTPORT( alternate-tcp-server-port )

Note:
P All parameters preceding this point can be specified
positionally.

```

The remove retail store entry command is used to remove store entries from the store information configuration table.

```

Job: B,I Pgm: B,I REXX: B,I Exec
Threadsaf: No

RMVRTLSTOR
STORENBR( retail-store-number )

```

Working with Retail Communications

This chapter gives you the information needed to work with retail communications using the functions and commands provided by RAIS. The RAIS LU0 support uses the Advanced Data Communication for Stores (ADCS) emulation when communicating with the retail controller. The RAIS TCP/IP support uses Retail Connectivity Option commands when communicating with the retail controller Retail Connectivity Option server. For host communications RAIS uses the host command processor (HCP) emulation. A detailed description is provided for each function and command.

Retail File Communications Configuration

LU0 Support: The LU0 support of RAIS uses ADCS emulation to send ADCS commands to the HCP program. The Retail Controller Description(CRTCTLRTL) and the Retail Device Description(CRTDEVRTL) are the RAIS configuration objects used for LU0. The RAIS location manager will use the Remote Location of the Retail Device to determine the LU0 location. See "Retail Controller Configuration Examples."

TCP Support: The TCP/IP support uses Retail Connectivity Option commands to communicate with the Retail Connectivity Option servers on the retail controller. See "Using Retail Connectivity Option server." for more information on the Retail Connectivity Option TCP server. The RAIS location manager will use the Locally defined TCP Host Name entries, or the Host Name entries returned from the Configured Domain Name server to determine the TCP location.

Using Retail Connectivity Option server

When the TCP/IP protocol is used for RAIS applications, the RAIS API communicates with the Retail Connectivity Option server on the 4690. The TCP server must be started manually or as a background job on the 4690.

ADXSRVTL.286 - 4690 TCP/IP Server

The 4690 Server provides both character and record mode file services for a host ISP (In Store Processor). The 4690's TCP/IP communications offering is required and must be running.

This program may be run in either "foreground" or "background", but would normally be started as a background task at IPL.

ADXSRVTL.286 expects a 4690 File Table file, C:\ADX_SDT1\ADXFTBLF.DAT, which is used to manage access to critical 4690 files. For more information on the configuration file see the program description for **ADXBFTBL.286** (the file table maintenance program).

Command Format

```

ADXSRVTL - TCP/IP Server
  ADXSRVTL
    -Pnnnn      HELP
                -H
                ?

  QUIET        STATIC      LOGERR
  -Q           -S          -E
  DEBUG
  -D

  -Ffilename   -Bnnnn     -Ix

  -Nn          -Mx
  
```

HELP : No Communications, just display command options.

and the message modes are:

QUIET : Error Messages only
 NORMAL : (Default) Some limited operational messages
 DEBUG : Extensive operational messages

other parameters

STATIC : Suppress the dynamic expansion of the file table.
 This will require that all files referenced be in the initial table.
 LOGERR : Enable 469 application event logging.

-Ffilename : To specify an alternate file table name.
 (C:\ADX_SDT1\ADXFTBLF.DAT is default)
 -Pnnnnn : TCP/IP TCPPORT to use (default is 2).
 -Q : alternative for "QUIET"
 -D : alternative for "DEBUG"
 -S : alternative for "STATIC"
 -E : alternative for "LOGERR"
 -B : maximum block size
 (default & maximum value is 16384)
 -H or ? : alternatives for "HELP"
 -Ix : x specifies the default PRS pipe for q_prs_read function
 -Nn : specifies that event number n is used when logging events (default is 64)
 -Mx : specifies that message group x is used when logging events (default is 'M')

The QUIET, DEBUG, and LOGERR modes can also be turned on and off from the in-store processor using the **posreset** command.

The server's operation can also be modified by the presence of the file \ADX_UDT1\RCORESET.CMD. This file can be created by another application, transmitted from the host, copied from a diskette, created using an editor, or any other technique for creating a small text file. If present, the file is assumed to contain an ASCII text string, not exceeding 100 characters. The server will check for the presence of this file approximately every 15 minutes. If the file is found, its contents (up to 100 characters) will be read and the file will be deleted. The data read will then be checked for the presence of key words indicating actions to be taken. Multiple key words can be specified at the same time. The following key words are checked for and processed in the order listed.

DEBUGON turns debug mode on. This only has meaning if the server is being executed in command mode.

DEBUGOFF turns debug mode on. This only has meaning if the server is being executed in command mode.

QUIETON turns quiet mode on.

QUIETOFF turns quiet mode off.

LOGON turns error logging on.

LOGOFF turns error logging off.

STOP terminates execution of the server.

RESTART causes the server to be restarted.

ADXBFTBL.286 - 4690 File Table Maintenance Program

ADXBFTBL provides creation and maintenance functions for the 4690 File Table file, **C:/ADX_SDT1/ADXFBLF.DAT**.

The 4690 File Table file is a 4690 resident file that defines 4690 files that are available to Retail Connectivity Option applications on the AS/400. The 4690 File Table file can contain references of up to 128 files. If this does not seem to be enough, it is likely that the use of wild-card character "?" in the file names should be used. This is described below. As a security feature if the server program is initiated with the **STATIC** or **-S** command line variable, access is limited to files defined in the 4690 File Table file when the server was loaded.

For RAIS polling applications, included in this file table is a timeout value. This time out value is the number of seconds the file is left open before the server will close the file. If an application will be keeping a file open for long periods of time and may have large delays where the file is not accessed, the file should be defined in the file table with an appropriate time out value. The timeout value is in seconds, and a value of 0 means it will not timeout. If no entry is specified for a file, the default timeout is 60 seconds.

The presence of a file's definition in this 4690 File Table file also makes record level data conversion possible when AS/400 applications access POS data. When a file is opened, its name is compared with those in the 4690 File Table file. If a match is found and the **O_NOXLATE** option was not used, the relative position of the 4690 File Table file entry identifies the record conversion format in the AS/400's format conversion file **/etc/formats.inp**. A successful match depends on several conditions being met.

The names must match exactly in length and characters used.

Upper and lower case characters are considered equivalent.

```
"FILE.DAT" will match "file.dat"
```

The characters "/" and "\" are considered equivalent.

```
"C:\ADX_IDT4\EAMIMOVE.DAT" matches  
"C:/ADX_IDT4/EAMIMOVE.DAT"
```

The character "?" used in the file name in the 4690 File Table file matches any character in the corresponding position of the name used in the open function. This wild card feature allows files with similar names and matching data formats to share a single entry in 4690 File Table file.

```
"FILE 1.DAT", "FILE 2.DAT", & "FILE 3.DAT  
all match  
"FILE???.DAT"
```

This utility is menu driven and allows the user to add, update, delete, and display entries in the 4690 File Table file.

Command Format

```
ADXBFTBL - 4690 File Table Maintenance  
ADXBFTBL file_table_file_name
```

The following information will be requested for records in the file.

```
Filename - Full Path/filename or Logical filename of  
the file.  
File Type - Is file keyed?  
(if no then direct or sequential assumed)  
Record Length - Keyed Files only (max of 5 8)  
Key Length - Keyed Files only  
Access Mode - { read only : write only : read/write }  
Time Out - Number of Seconds to keep the file open  
without activity
```

Example: ADXBFTBL

This would be the command sequence to perform maintenance on the production configuration file.

Notes:

1. The filename is limited to 24 characters, however Logical names are supported for longer file names.
2. The file name must agree on a character by character basis with the name used to open the file from the AS/400 system. However alphabetic characters will match regardless of whether they are upper or lower case. The characters "/" and "\" are also considered synonymous.
3. File names can contain the wild card character "?". When comparing file names, searching for their table entry, any character will match a "?" used in the file name in the file table.
4. The timeout is entered as seconds (a value of 0 will suppress the time out function).
5. Dummy or reserved entries are allowed to hold a place in the table. Null entries (created by delete functions) are also allowed.
6. The file table is read into memory once during server initiation. Therefore, for the server to recognize modifications to the file table, the server must be stopped and restarted.

File Structures

Several retail file structures are used in file conversions. They include the following:

Keyed Files on a Retail Controller

Keyed 4690 or 4684 retail controller files are composed of fixed-length records that are accessed using a key. These records are blocked in 512-byte sectors. The first 4 bytes of each sector are used for forward and backward chaining pointers, and the remaining 508 bytes are used for record storage. A sector can contain multiple records, but a record cannot span sectors; therefore, the maximum record length of a 4690 or 4684 keyed file is 508 bytes. The record key can be as many bytes of the record as desired. The first sector of each file is reserved for parameters describing the file and cannot be used for data.

Keyed 3651 and 3684 retail controller files are composed of either fixed length or variable length records that are accessed using a key. These records are blocked in 256-byte sectors, and records cannot span sectors. The first 2 bytes of each sector are reserved; therefore, the maximum length of a fixed length record is 254 bytes. Variable length records have a 1-byte length indicator (in addition to the chaining pointer) at the start of each record; therefore, the maximum length of a variable length record is 253 bytes. All sectors of 3651 and 3684 retail controller files can contain data because the file description parameters are kept in a separate file (VTOC).

Records in keyed files are accessed by a technique that uses key folding and randomizing division. For details of this process, see the programmer's guide for the appropriate retail point-of-sale controller.

Sequential Files on a Retail Controller

Sequential 4690 retail controller files are composed of variable length records that are accessed consecutively from first to last. A record in a sequential file consists of one or more data fields, each delimited by commas. String data fields are delimited by apostrophes and colons in addition to the commas. The records are delimited by carriage-return and line-feed characters. Sequential file records can be up to 32KB (KB equals 1024 bytes) in length and can span sectors.

Sequential 3651 and 3684 retail controller files are composed of fixed- or variable-length records. Fixed-length records can be up to 256 bytes long. Variable-length records have a 1-byte length indicator at the start of each record, therefore

they can be up to 255 bytes long. Neither fixed- nor variable-length records can span sectors, but a sector can contain multiple records.

Sequential 4684 retail controller files are similar to 3651 and 3684 files except that their sector size is 512 bytes and variable-length records have a 2-byte length indicator.

Retail File Conversion

Retail file conversion converts IBM point-of-sale files into formats easily accessible by your application code on the AS/400 system. This means that all ASCII character data is converted to EBCDIC and all numeric data is converted to either packed decimal or character format.

You have the option of defining, in retail controller terms, the format of incoming point-of-sale files and, to a limited degree, the format of the converted output file. These definitions are contained in named conversion templates.

You can translate IBM point-of-sale data into AS/400 format by specifying the proper template name during the file conversion process.

This data can be inserted into an AS/400 relational database collection by specifying a template that was created for that specific collection.

You also have the capability of requesting that the file conversion routines generate data description specification (DDS) source statements to describe the output file generated by the conversion as well as RPG/400 and COBOL/400 source statements for those languages.

Work with Retail Conversion Templates

Select option 3 (Work with retail conversion templates) from the Work with Retail File Conversion menu, or use the WRKRTLTP command to add, change, or delete retail conversion templates. Example templates are available for the two most common 4690 point-of-sale applications, IBM General Sales Application (GSA) and IBM Supermarket application. Each template can be made up of multiple template records. Each template record is made up of multiple fields.

Note: If you are using the IBM 4690 Chain Drug Sales Application be sure to read "Chain Sales Special Considerations" on page 2-20. This display provides maintenance for retail conversion templates. The user can maintain templates provided with this application, or create new ones.

```

Work with Retail Conversion Templates

Position to . . . Starting characters of template

Type options, press Enter.
 2=Change  3=Copy  4=Delete  5=Display  6=Print  7=Rename
 8=Work with records  9=Create source statements

Retail
Option  Template      Text
-      EALACCTC      GSA Accounting
-      EALDEPTC      GSA Department Totals
-      EALDMCTL      GSA Delayed Maintenance Control
-      EALDMERR      GSA Delayed Maintenance Error file
-      EALDPLST      GSA Department Totals List
-      EALIMLST      GSA Item Movement List
-      EALIMOV      GSA Item Movement
-      EALPERFC      GSA Operator Performance
-      EALTPROD      GSA Terminal Productivity
-      EALTRANS      GSA Transaction Summary Log
-      EAMACCTC      SMA Accounting Totals

F3=Exit  F6=Create template  F12=Cancel

```

```

Change Retail Conversion Template

Retail template . . . . . : EALTRANS

Type choices, press Enter.

Retail controller type . . . . . 469  469 , 4684, 3651, 3684
Retail file organization . . . . . 1  1=Sequential
                                         2=Keyed
                                         3=Random
                                         4=Direct
                                         4=512
Retail record length . . . . .
Variable length fields . . . . . Y  Y=Yes, N=No
String file . . . . . Y  Y=Yes, N=No
Format selector type . . . . . 2  1=Single format
                                         2=Fixed location
                                         3=Field number
                                         4=Relative record number

For choice 2, 3 or 4:
Location or number . . . . . 1  1-512
Format selector length . . . . . 1  1-5

F3=Exit  F12=Cancel  F16=Work with records

Bottom

```

Change Retail Conversion Template: Select option 2 (Change) from the Work with Retail Conversion Templates display to change header information describing a template.

See "Retail Conversion Template Field Descriptions" on page 2-10 for an explanation of the data to be entered at each prompt.

You can page up (ROLLDOWN) to access the first change display prior to pressing the Enter key.

```

Change Retail Conversion Template

Retail template . . . . . : EALTRANS

Type choices, press Enter.

Text . . . . . GSA Transaction Summary Log

Retail file . . . . . GSATRANName
Library . . . . . LIBL Name, CURLIB, LIBL

AS/4 file . . . . . EALTRANName
Library . . . . . LIBL Name, CURLIB, LIBL

DB2 Collection . . . . . Name

Include store ID and date
in output data record . . . Y  Y=Yes, N=No

More...

F3=Exit  F12=Cancel  F16=Work with records

```

Copy Retail Conversion Templates: Select option 3 (Copy) from the Work with Retail Conversion Templates display to create a new template from an existing template. All records and fields will be copied.

```

Copy Retail Conversion Templates

Type To Retail Template name, press Enter.

From Retail  To Retail
Template     Template
EALTRANS    _____

```

See "Retail Conversion Template Field Descriptions" on page 2-10 for an explanation of the data to be entered at each prompt.

Enter names for *To Retail* templates. These will be the names assigned to the copies. All retail conversion templates must have unique names. The names must conform to rules for simple names on the AS/400 system and they cannot begin with the letter Q.

Enter the information you want to change and then press the Enter key. If the data you want to change is not displayed, page down (ROLLUP) for the next display. The name of the retail template cannot be changed using the change function.

Confirm Delete of Retail Conversion Templates: Select option 4 (Delete) from the Work with Retail Conversion Templates display to confirm deletion of retail conversion templates.

The second part of the change template display provides global information about the retail controller file that is necessary to properly convert it. If variable length fields (NO) is specified, the file must have fixed-length fields. A string file is a special type of transaction file that has multiple record types of a single sales transaction in a long string.

```

Confirm Delete of Retail Conversion Templates

Press Enter to confirm your choices for 4=Delete.
Press F12 to return to change your choices.

Retail
Option  Template      Text
4      EALTRANS      GSA Transaction Summary Log

```

Press the Enter key to remove all the selected templates. Press F12 (Cancel) to return to the Work with Retail Templates display without deleting the selected templates.

Display Retail Conversion Template: Select option 5 (Display) from the Work with Retail Conversion Templates display to see header information describing a template.

```

Display Retail Conversion Template

Retail template . . . . . : EALTRANS
Text . . . . . : GSA Transaction Summary Log

Retail file . . . . . : GSATRANS
Library . . . . . : LIBL

AS/4 file . . . . . : EALTRANS
Library . . . . . : LIBL

DB2 Collection . . . . . :

Include store ID and date
in output data record . . . : Y

Press Enter to continue.
More...

F3=Exit F12=Cancel F16=Work with records

```

See “Retail Conversion Template Field Descriptions” on page 2-10 for an explanation of the data shown on this display.

The second part of the display template display provides global information about the retail controller file necessary to properly convert it. You must page down (ROLLUP) to access the second retail display prior to pressing the Enter key.

```

Display Retail Conversion Template

Retail template . . . . . : EALTRANS

Retail controller type . . . . : 469
Retail file organization . . . : 1
                                1=Sequential
                                2=Keyed
                                3=Random
                                4=Direct

Retail record length . . . . . :
Variable length fields . . . . : Y
String file . . . . . : Y
Format selector type . . . . . : 2
                                1=Single format
                                2=Fixed location
                                3=Field number
                                4=Relative record number

Location or number . . . . . : 1
Format selector length . . . . : 1

Press Enter to continue.
Bottom

F3=Exit F12=Cancel F16=Work with records

```

See “Retail Conversion Template Field Descriptions” on page 2-10 for an explanation of the data to be entered at each prompt.

You can again access the first display by paging up (ROLLDOWN).

Rename Retail Conversion Templates: Select option 7 (Rename) from the Work with Retail Conversion Templates display to change the name of an existing template.

```

Rename Retail Conversion Templates

To rename retail template, type New Name, press Enter.

Retail      New
Template    Name
EALTRANS   _____

```

Enter names for *New Name* to be assigned to the templates. All retail conversion templates must have unique names. The names must conform to rules for simple names on the AS/400 system and they cannot begin with the letter Q.

Create Retail Conversion Template: Select F6 (Create template) from the Work with Retail File Conversion Template display to add information to create a template.

```

Create Retail Conversion Template

Type choices, press Enter.

Retail template . . . . . Name
Text . . . . .

Retail file . . . . . Name
Library . . . . . Name, CURLIB, LIBL

AS/4 file . . . . . Name
Library . . . . . Name, CURLIB, LIBL

DB2 Collection . . . . . Name

Include store ID and date
in output data record . . . Y=Yes, N=No

More...

F3=Exit F12=Cancel

```

See “Retail Conversion Template Field Descriptions” on page 2-10 for an explanation of the data to be entered at each prompt.

Enter the data necessary for creating your conversion template. You must page down (ROLLUP) to access the second create display prior to pressing the Enter key.

This is the second part of the Create Template display.

```

Create Retail Conversion Template

Type choices, press Enter.

Retail controller type . . . . 469 , 4684, 3651, 3684
Retail file organization . . . 1=Sequential
                                2=Keyed
                                3=Random
                                4=Direct

Retail record length . . . . . 4=512
Variable length fields . . . . Y=Yes, N=No
String file . . . . . Y=Yes, N=No
Format selection type . . . . 1=Single format
                                2=Fixed location
                                3=Field number
                                4=Relative record number

For choices 2, 3 or 4:
Location or number . . . . . 1-512
Format selector length . . . . 1-5

Bottom

F3=Exit F12=Cancel

```

Finish entering your data and then press the Enter key. You can again access the first create display by paging up (ROLLDOWN).

Retail Conversion Template Field Descriptions:

Use the following descriptions to help you in creating and changing templates and in interpreting template displays.

Retail template

This is the name associated with a template. It is used to identify this template when you request conversion to or from retail files. Retail template names must conform to AS/400 rules for simple names and they cannot begin with the letter Q.

Text

This is descriptive text to help you in identifying the purpose of this template.

Retail file/library

This is a default qualified file name to be associated with the template. It represents the retail format data that is input to or output from a conversion. This file is used if you specify the special value *RTLTP in one of the following:

CVTFMRRLF command on parameter FROMFILE
CVTTORRLF command on parameter TOFILE
Convert from retail file format display prompt Input file name
Convert to retail file format display prompt Output file name

AS/400 file/library

This qualified file name is used in the same manner as retail file/library names. It represents the AS/400 data format file in a conversion. It is used by the Polling Application Conversion when a DB2 Collection is not specified for this template. See Chapter 5, "RAIS Retail Polling Application" on page 5-1 for additional information on the polling application. It is also used if you specify the special value *RTLTP in one of the following:

CVTFMRRLF command on parameter TOFILE
CVTTORRLF command on parameter FROMFILE
Convert from retail file display prompt Output file name
Convert to retail file display prompt Input file name

DB2 Collection

This can be the name of a pre-defined relational database collection that will be used for this template. This also could be the name of an AS/400 library that holds AS/400 physical files to store the retail data. The record and field template support will use this name as the library reference when creating the record definitions and field definitions for this template. The polling application conversion will use this field to insert the converted retail data into the relational database collection or library.

Include store ID and date in output data record

This function inserts an extra 10 bytes of data into your output file on a conversion from retail format to AS/400 format. It removes the 10 bytes of data on a conversion from AS/400 format to retail format. The 10 bytes of data

are located immediately following the record format selectors of the AS/400 format file, in the following format:

IIIIIIIIII where:

For Conversion Commands:

IIII = Bytes 2 through 5 of the name of the retail file member being converted.
DDDDDD = Represents the same data as bytes 6 through 10 of the retail file member being converted. Bytes 6 through 10 of the retail file member name are assumed to contain a date in Julian format. The format of this date is the same as that specified by system value QDATFMT on your AS/400 system.

Note: This function will work properly only if the input retail file members are named in accordance with the rules established by the RCVRLF command TOMBR(*STORE) option.

Y=YES means to start this function

N=NO means *not* to start this function

Retail controller type

This entry is restricted to the four values 4690, 4684, 3651, and 3684. It is imperative that this value reflect the type of retail controller from which the retail file was received or to which the retail file is sent. This value defines the structure of the various file types to the conversion code.

Note: When configuring the template to use the DB2 Collection support, this must be set to 4690.

Retail file organization

This entry is restricted to the four values 1, 2, 3, and 4 where:

1 = Sequential
2 = Keyed
3 = Random
4 = Direct

This value instructs the conversion code how to locate and build retail records. String files should be declared as sequential.

Note: When using the DB2 Collection reference support, this may not be set to 2 = Keyed.

Retail record length

This entry is required for retail files with random organization. It should indicate the actual physical length of the retail record from the first data byte up to and including the ending carriage return and line feed.

Variable length fields

This entry indicates to the conversion code whether the fields in the retail format are of fixed or variable length. If variable length fields are indicated it is assumed that commas are used as delimiters between fields. For string files following the structure of the 4690 General Sales Application, 4690 Supermarket Application, or the 4690 Store Management Application, this field should be

coded as YES. For string files following the structure of the 4690 Chain Drug Sales Application, this field should be coded as NO.

For Direct file organization, if variable length fields are indicated and the defined Retail record length is at least 2 Greater than the calculated length, a CRLF(0D0A) delimiter will be set at the end of the Record.

String file

This entry indicates to the conversion code that the special string format is used for this retail file. This entry should be coded as YES for string file only.

Format selector type

This entry is limited to four entries (1, 2, 3, or 4) with the following meanings:

- 1 All records within the retail file are of the same format.
- 2 There are multiple record formats within the retail file. The different formats are identified by a unique bit pattern per format. The unique bit pattern is always located at the same relative position within each record format.
- 3 There are multiple record formats within the retail file. The different formats are identified by a unique bit pattern per format. The unique bit pattern is always contained within the same field number within each record format. This type is valid only for Retail file organization = 1 (Sequential) with variable length fields specified as Yes.
- 4 There are multiple record formats within the retail file. The different formats are identified by the relative record of the record within the file.

Location or number

For files with multiple record formats specify the following:

For format selector type 2 (Fixed location) specify the starting byte number for the unique bit pattern (the first byte is 1, not 0).

For format selector type 3 (Field number) specify the number of the field for the unique bit pattern (the first field is 1, not 0).

For format selector type 4 (Relative record number) specify any number.

Format selector length

Specify this entry whenever you have specified a value for the location or number. This is the number of bytes to be used for unique bit patterns for format selector types 2 and 3. It is the number of digits to be specified for format selector type 4. This entry must be in the range of 1 to 5.

Exit Work with Retail Template: The Exit Work with Retail Template display appears when you exit retail conversion template record maintenance. Maintenance is done in a work file so that you will have the option to disregard changes.

F3 (Exit) or F12 (Cancel) from the Work with Records in Conversion Template display selects this display.

```

                                Exit Work with Retail Template
Retail template . . . . . : EALTRANS
Type choices, press Enter.
Option . . . . . _ 1=Save changes and exit
                                     2=Exit without saving changes
                                     3=Resume work with records
For choices 1 or 2:
Print template output . . . . . _ Y=Yes, N=No
Create source statements . . . . . _ Y=Yes, N=No

F12=Cancel

```

Option 1 (Save changes and exit)

This option saves your changes and returns to the Work with Retail Template display.

Option 2 (Exit without saving changes)

This option exits the Work with Retail Templates display and discards changes you have made.

Option 3 (Resume work with records)

This option (or F12) returns you to the Work with Template Records display.

You have the option of causing a formatted print of the template you just worked on to be created or to create source statements when exiting. Enter a Y in the appropriate field for this function.

Work with Records in Retail Template

Option 8 (Work with records) from the Work with Retail Conversion Templates display, or F16 from the Change Retail Conversion Template display or the Display Retail Conversion Template display accesses this display.

Use this display to provide maintenance for the records in a specific template.

```

Work with Records in Retail Template
Retail template . . . . . : EALTRANS   Maximum record length . . . . . : 132
Position to . . . . . Starting characters of record
Type options, press Enter.
  2=Change  3=Copy  4=Remove  5=Display  7=Rename  8=Work with fields

Record
Opt Record Selector Text
- CSHDRWCHG 4 Cash Drawer Change
- DATAENTRY 9 Data Entry
- DATAMAIN 18 Data Maintenance
- DISCOUNT 1 Discount
- HEADER Header
- HEADEREXT 1 Header Extension
- LOG 16 Log Time Analysis
- MERCH 6 Merchandise/non-Merchandise
- MERCHEXT 7 Merchandise/non-Merchandise Extension
- MISC 8 Miscellaneous

F3=Exit F6=Add record F1 =Update DB2 Information
F12=Cancel

```

Multiple record types are supported to handle point-of-sale summary files. The record selector is the expanded hex-character identifier or relative-record-number selector.

Select the option number for the function you want to run on the Work with Records in Retail Template display and press the Enter key.

Function F6 (Add record) is available only if the format selector type is specified as 2, 3, or 4 in the retail conversion template and there are less than 100 record types defined within this template.

Function F10 (Update DB2 Information) is available only if the DB2 Collection name is specified in the retail conversion template and there are less than 100 record types defined within this template.

Add Retail Template Record: Press F6 (Add record) from the Work with Records in Retail Template display to add a retail controller record definition to a conversion template. A retail file can have up to 50 different record types in it. Each record must be identified by a unique record format selector or be indicated as a Y value in the *Catchall record* prompt. Each record will have multiple fields also.

```

Add Retail Template Record
Retail template . . . . . : EALTRANS
Type choices, press Enter.
Record . . . . . Name
Record format selector . . . . . Hexadecimal number
Catchall record . . . . . Y=Yes, N=No
Omit record type . . . . . Y=Yes, N=No
DB2 Table . . . . . Name
AS/4 exit program . . . . . Name
Library . . . . . Name, CURLIB, LIBL
Text . . . . .

F3=Exit F12=Cancel

```

See "Retail Conversion Template Record Field Descriptions" on page 2-13 for an explanation of the data to be entered on this display.

Change Retail Template Record: Select option 2 (Change) from the Work with Records in Retail Template display to change the definitions of an existing retail conversion template record. Record names cannot be changed using this function. The record format selector provides information about how to identify the specific records in a multiple record type file. The omit option can be used to drop unwanted record types from the converted output file. The record format selector is used to identify the record type.

```

Change Retail Template Record
Retail template . . . . . : EALTRANS
Record . . . . . : DISCOUNT
Type choices, press Enter.
Record format selector . . . . . 1 Hexadecimal number
Catchall record . . . . . N Y=Yes, N=No
Omit record type . . . . . N Y=Yes, N=No
DB2 Table . . . . . Name
AS/4 exit program . . . . . Name
Library . . . . . Name, CURLIB, LIBL
Text . . . . . Discount

F3=Exit F12=Cancel F16=Work with fields

```

Type the data you want to change, and then press the Enter key. See "Retail Conversion Template Record Field Descriptions" on page 2-13 for an explanation of the data on this display.

Copy Records from Conversion Template: Select option 3 (Copy) from the Work with Records in Retail Template display to copy a specific record to a new record in the same template. All the fields in the record are copied as well.

This option is valid only when format selector type is specified as 2, 3 or 4 for the retail conversion template and there are less than 100 record types defined for the template.

```

Copy Records from Conversion Template
Retail template . . . . . : EALTRANS
Type To Record name and To Record Selector, press Enter.
From Record To Record To Record Selector
CSHDRWCHG . . . . . 4

```

Type the values in the *To Record* and *To Record Selector* fields for the copies and press the Enter key. Record names

and record selectors must be unique within a given conversion template.

The entries in the *From Record* field name the retail conversion template records to be copied. Enter the new record name in the *To Record* field to be assigned to the copy of the record. Record names must conform to the rules for simple names on the AS/400 system and must be unique within a given retail conversion template. Enter a new record selector for each copy. A record selector must consist of one to five (format selector length from template header), valid hexadecimal digits, or a valid relative record number, and must be unique within a given retail conversion template. Valid hexadecimal digits are 0 through 9 and A through F.

Confirm Remove of Template Records: Select option 4 (Delete) from the Work with Records in Retail Template display to confirm the deletion of template records from a retail conversion template. This function deletes the records and all the fields in it.

```

Confirm Remove of Template Records

Retail template . . . . . : EALTRANS

Press Enter to confirm your choices for 4=Remove.
Press F12 to return to change your choices.

      Record
Opt Record Selector Text
 4 CSHDRWCHG 4 Cash Drawer Change

```

Display Retail Template Record: Select option 5 (Display) from the Work with Records in Retail Template display to see information regarding record identification and omission in multiple record format retail controller files.

```

Display Retail Template Record

Retail template . . . . . : EALTRANS
Record . . . . . : CSHDRWCHG

Record format selector . . . . . : 4 Hexadecimal number

Catch All record . . . . . : N

Omit record type . . . . . : N

DB2 Table . . . . . :

AS/4 exit program . . . . . :
Library . . . . . :

Text . . . . . : Cash Drawer Change

```

See "Retail Conversion Template Record Field Descriptions" for an explanation of the data on this display.

Rename Records in Retail Template: Select option 7 (Rename) from the Work with Records in Retail Template display to rename a record.

```

Rename Records in Retail Template

Retail template . . . . . : EALTRANS

To rename record, type New Name, press Enter.

Record      New Name
CSHDRWCHG  _____

```

Enter new names for template records and then press the Enter key. Template record names must be unique within a conversion template.

Updating DB2 Information: Press F10 (Update DB2 Information) from the Work with Records in Retail Template display to update the DB2 Table information associated with the record definition to a conversion template. This should be used when any changes are made to the Relational Database Collection specified in the conversion template DB2 Collection. If this is not done the Table relationships will not be reflected in the Templates.

Retail Conversion Template Record Field Descriptions

Retail template

This is the name of the retail template within which you are working.

Record

This is the name you want assigned to this record format. It must conform to the AS/400 rules for simple names. Record names must be unique within a given retail conversion template.

Record format selector

This field is required only if multiple record types (format selector types 2, 3, or 4) exist in this retail conversion template. Record format selectors are composed of either hexadecimal digits or a relative record number based on the format selector type. For format selector types 2 and 3, a record format selector consists of twice the number of hexadecimal digits as the length specified previously for format selector length. Valid hexadecimal digits are 0 through 9 and A through F. For format selector type 4, a record format selector consists of a relative record number, whose number of digits is the same as the format selector length specified previously. Record format selectors must be unique within a given retail conversion template.

If this record is a catchall record, you are not required to specify a record format selector.

Catchall record

You can specify Yes or No for this entry. A catchall record is a record format that will be used by the file translation system to translate all records that do not meet the explicit format selection requirements of any other record format in this template. An explicit record format definition *always* takes precedence over a catchall record. You may have only one catchall record in a retail conversion template.

Omit record type

This entry instructs the retail file conversion system whether to convert records identified by this record selector. If YES is specified, records with this record selector will be omitted (not converted). If NO is specified, records with this record selector will not be omitted (they will be converted).

DB2 Table

This entry instructs the polling application conversion to insert the converted records identified by this record selector into this DB2 Table that resides in the template defined relational database collection or DB2 Collection.

AS/400 exit program/library

If this field is specified, the polling application will call this program after converting the retail data. If no fields are defined as part of this template record, the unconverted data will be sent to the exit program. This parameter is used by the polling application. See "Polling Application Exit Program Considerations" on page 5-7 for information on how to set up your exit program.

Text

This is descriptive information to help you in determining what retail record format this conversion template record pertains to.

Work with Fields in Template Record

Select option 8 (Work with fields) from the Work with Records in Retail Template display to see retail and AS/400 record layouts side by side.

Initial creation of fields for DB2 Table defined records.:

For a DB2 Collection defined template, the fields of any Record are predefined with the physical file field definitions of the specified table. The user then is able to add, delete or change the fields to reflect the actual conversion processing that is needed. For this reason, the DB2 Collection must be defined before a corresponding template can be created.

```

Work with Fields in Template Record

Retail template . . . . . : EALTRANS
Record . . . . . : CSHDRWCHG
Record format selector . . . : 4 Hexadecimal number

Type options, press Enter.
 2=Change 3=Copy 4=Remove 5=Display

-----Retail Record-----AS/4 Record-----
Field Data Field From Field Data Dec
Opt Sequence Type Size Field Position Size Type Pos
          RECID##GEN 1 1 Copy
          DATE##GEN 2 6 Right
          STORE##GEN 8 4 Right
- 1 Packed 1 STRINGTY 12 2 Packed
- 2 Packed 1 TEND 1 14 2 Packed
- 3 Packed 4 AMTT 1 16 5 Packed
- 4 Packed 1 TEND 2 21 2 Packed
- 5 Packed 4 AMTT 2 23 5 Packed

F3=Exit F5=Refresh F6=Add field F12=Cancel
F17=Top F18=Bottom F2 =Renumbr
    
```

The definitions for the *Data Type* field are:

Retail Record:

- ASCII** ASCII character field
- Packed** retail unsigned packed
- Reverse** 4690 or 4684 integer (inverted)
- Integer** 3651 or 3684 integer
- Flag** Flag byte
- Drop** Drop this field
- Item** Special item code field
- Ref** DB2 Collection referential field

AS/400 Record:

- Right** EBCDIC character (right adjusted)
- Left** EBCDIC character (left adjusted)
- Packed** AS/400 packed
- String** 8 bytes of string data
- Copy** Copy image field
- Fill** AS/400 pad field
- Item** Special item code field
- Binary** DB2 Collection binary data field
- Zoned** DB2 Collection zoned data field
- Date** DB2 Collection date field
- Time** DB2 Collection time field
- Timstmp** DB2 Collection time stamp field
- Float** DB2 Collection floating point data field

Change Field in Template Record: Select option 2 (Change) from the Work with Fields in Template Record display to change how the point-of-sale data fields will convert to AS/400 format.

```

Change Field in Template Record

Retail template . . . . . : EALTRANS
Record . . . . . : CSHDRWCHG
Record format selector . . . : 4 Hexadecimal number

Type choices, press Enter.

Field sequence . . . . . 1 1-99999
Retail data type . . . . . 2 1=ASCII data, 2=Packed 469
                                     3=Reverse integer 469
                                     4=Integer 36XX, 5=Flag byte,
                                     6=Drop field, 7=Item code

Maximum field size . . . . . 1 1-256
Multiple fields . . . . . N Y=Yes, N=No
AS/4 data type . . . . . 3 1=EBCDIC right, 2=EBCDIC left
                                     3=Packed, 4=String, 5=Copy
                                     6=Pad output, 7=Item code

Field . . . . . STRINGTYName
Decimal positions . . . . . -9 (AS/4 packed only)
Field option . . . . .

Action field . . . . .
F3=Exit F12=Cancel
    
```

Each field is described in point-of-sale and AS/400 data format.

The field names are used in source statements created for RPG, DDS, and COBOL.

Type the data to be changed, and then press the Enter key.

See "Fields within Retail Conversion Template Records Descriptions" on page 2-15 for an explanation of the data on this display.

Confirm Remove of Fields: Select option 4 (Remove) from the Work with Fields in Template Record display to confirm your request to remove field definitions from this template record.

```

Confirm Remove of Fields

Retail template . . . . . : EALTRANS
Record . . . . . : CSHDRWCHG
Record format selector . . . . : 4 Hexadecimal number

Press Enter to confirm your choices for 4=Remove.
Press F12 to return to change your choices.

-----Retail Record-----AS/4 Record-----
Field Data Field From Field Data Dec
Opt Sequence Type Size Field Position Size Type Pos
4 1 Packed 1 STRINGTY 12 2 Packed
  
```

Add Field in Template Record: Press F6 (Add field) from the Work with Fields in Template Record display to add data fields to a template record.

```

Add Field in Template Record

Retail template . . . . . : EALTRANS
Record . . . . . : CSHDRWCHG
Record format selector . . . . : 4 Hexadecimal number

Type choices, press Enter.

Field sequence . . . . . 1-99999
Retail data type . . . . . 1=ASCII data, 2=Packed 469
                          3=Reverse integer 469
                          4=Integer 36XX, 5=Flag byte,
                          6=Drop field, 7=Item code

Maximum field size . . . . . 1-256
Multiple fields . . . . . N Y=Yes, N=No
AS/4 data type . . . . . 1=EBCDIC right, 2=EBCDIC left
                          3=Packed, 4=String, 5=Copy
                          6=Pad output, 7=Item code

Field . . . . . FLD 1 Name
Decimal positions . . . . . -9 (AS/4 packed only)
Field option . . . . .

Action field . . . . .

F3=Exit F12=Cancel
  
```

Type the data necessary to define the field, and then press the Enter key.

See “Fields within Retail Conversion Template Records Descriptions” for an explanation of the data to be entered on this display.

Display Field in Template Record: Select option 5 (Display) on either of the Work with Fields in Template Record displays to see the specific information used by the conversion routines to convert specific fields from retail controller format to AS/400 formats.

```

Display Field in Template Record

Retail template . . . . . : EALTRANS
Record . . . . . : CSHDRWCHG
Record format selector . . . . : 4 Hexadecimal number

Field sequence . . . . . 1
Retail data type . . . . . 2 1=ASCII data, 2=Packed 469
                          3=Reverse integer 469
                          4=Integer 36XX, 5=Flag byte,
                          6=Drop field, 7=Item code

Maximum field size . . . . . 1
Multiple fields . . . . . N Y=Yes, N=No
AS/4 data type . . . . . 3 1=EBCDIC right, 2=EBCDIC left
                          3=Packed, 4=String, 5=Copy
                          6=Pad output, 7=Item code

Field . . . . . STRINGTY
Decimal positions . . . . .
Field option . . . . .
Action field . . . . .

Press Enter to continue.

F3=Exit F12=Cancel
  
```

See “Fields within Retail Conversion Template Records Descriptions” for an explanation of the data shown on this display.

Copy Fields in Template Record: Select option 3 (Copy) from the Work with Fields in Template Record display to create new fields from a copy of existing fields.

```

Copy Fields in Template Record

Retail template . . . . . : EALTRANS
Record . . . . . : CSHDRWCHG
Record format selector . . . . : 4 Hexadecimal number

Type To Field Sequence and To Field name, press Enter.

From To
Field Field Data Field From To
Sequence Sequence Type Size Field Field
1 Packed 1 STRINGTY FLD 1
  
```

See “Field Sequence” and “Field” under “Fields within Retail Conversion Template Records Descriptions” for an explanation of the data to be entered in these fields.

Fields within Retail Conversion Template Records Descriptions

Retail template

This is the name of the retail conversion template you are working with.

Record

This is the name of the retail conversion template record you are working with.

Record format selector

This is the retail record format selector of the record you are working with.

Field sequence

This numeric field is used to control the order which the conversion system uses to interpret fields in files. Valid entries are the numbers from 1 to 99999 inclusive. Duplicate sequence numbers are not permitted. To add a field to the end of the record, use a sequence number higher than the last sequence number already defined. To add

a field at a position other than the end, use a sequence number that will fall numerically in the position desired. To move a field from one position in the list to another position, use the change function and change the sequence number to fall numerically in the correct position in the list. If there are no remaining unused sequence numbers between two fields (where you need to add a field or move a field) use F20 (Renumber) on the Work with Fields in Template Record display to insert space.

Retail data type

Enter a number from 1 through 7 to indicate the correct representation of the data in retail format. The meanings of the values 1 through 7 are as follows:

- 0 (Output Only)** When the field definitions for a DB2 Table defined record are first created, all Retail Data types are set to a 0 value. These may be change to represent the actual data type of the retail data. For the field entries that remain 0, there will be no retail processing by the polling application. Also, if a field is added, it will be set to 0.
- 1** ASCII. Data in this field is in character format.
- 2** Packed 4690. Data in this field is coded as unsigned packed decimal.
- 3** Reverse integer 4690. Data in this field is coded as binary integer numeric data. This data type should be used for binary integer data on the 4684 also.
- 4** Integer 36XX. Data in this field is coded as binary integer data on the 3651 and 3684.
- 5** Flag byte. Data in this field is to be interpreted as 8 bit sensitive switches within a single byte.
- 6** Drop field. Data within this field is to be dropped when converting from retail files.
- 7** Item code. Data within this field is in special item code format. This should only be used for 4690 GSA and 4690 SMA. For more information, see "Special item code handling for data types" on page 2-19.
- 8 (Not shown)** DB2 Collection referential field. This is set at the initial field description creation for the DB2 Table defined records. This field may not be displayed, changed or deleted. The field type is only displayed on the "Work with Fields in Template Record" screen with retail data type specified as Ref. This field represents a key field that is referentially constrained on some other database table and will be set to the referential keyed data during conversion.

For more information on retail data types, see "Data type field size limitations" on page 2-18.

Maximum field size

This is the maximum size in bytes of this field in retail format. For files with fixed-length fields, this entry should

be the exact field size in bytes. For files with variable-length fields, this entry should be the largest number of bytes that the field can occupy. Space will be allocated in the AS/400 record format to accommodate this maximum field size.

For more information on maximum field size, see "Data type field size limitations" on page 2-18.

Multiple fields

This indicates a retail string delimited field which contains more than 1 field value within the delimiters. The Maximum field size should be set to the exact size of the retail field. This indicator should not be set for the last field within the multiple delimited field.

AS/400 data type

Enter a number from 1 through 7 to indicate the correct representation of data in AS/400 format. The meanings of values 1 through 7 are as follows:

- 0** DB2 Table field type. When the field definitions for a DB2 Table defined record are first created, the AS/400 data type is retrieved from the AS/400 Physical file definition. The data types that are unchangeable are set to 0 in the field definition. The following is a list of DB2 Table record field data types that are unchangeable(0):

- binary** integer or short integer defined data
- float** real, float, or double defined data
- zoned** numeric(x,y) defined data
- date** date defined data
- time** time defined data
- timestamp** time stamp defined data

Note: For the DB2 Table record field data that are changeable, the following values may be used:

- alphanumeric** AS/400 data type can be 1,2,4,or 7
- packed or decimal(x,y)** AS/400 data type must be 3
- alphabetic** AS/400 data type can be 1,2,or 4
- numeric (character)** AS/400 data type can be 1,2,or 4

- 1** EBCDIC right. Data in this field is in character format. The data has been right-justified by truncating trailing blanks. This data can also be used for zoned decimal data if the retail data type is number 2, 3, or 4.
- 2** EBCDIC left. Data in this field is in character format. The data has been left-justified by removing leading zeros and leading blanks.
- 3** Packed. Data in this field is in AS/400 packed decimal format.

4 String.

Non-DB2 Collection specified template

Data in this field is a string of either 8, 12, or 24 bytes depending on input data type and size. Each byte contains a character value of 0 or 1 (X'F0' or X'F1') representing the bit string of 1, 2, or 3 bytes.

DB2 Collection specified template

Data in this field represents a string of either 8, 16, or 32 bytes depending on input data type and size. Each byte contains a character value of 0 or 1 (X'F0' or X'F1') representing the bit string of 1, 2, or 3 bytes. The string data type *should not* be used on the actual DB2 Table field entry. A new entry should be included with a sequence number directly ahead of the list of Zoned, or 1 character data. The template support will then make the following 8, 16 or 32 bytes unusable. The fields in the DB2 Table must be placed in the order they are represented in the 4690 bit stream.

Note: A field defined with a maximum length of 1 or 2 bytes will expand to an 8-byte flag. A field defined with a maximum length of 3 bytes will expand to a 16-byte flag. A field defined with a maximum length of 4 bytes will expand to a 32-byte flag.

- 5 Copy as is. Data in this field is in exactly the same representation as in the retail format file. The bytes have been copied with no conversion taking place.
- 6 Pad output. There is no data in this field. This data type is used to cause the conversion system to insert blank space in the AS/400 format for your applications to use.
- 7 Item code. Data in this field is in special item code format. This should only be used for 4690 GSA or 4690 SMA. For more information, see "Special item code handling for data types" on page 2-19.

For more information on data type field size limitations, see "Data type field size limitations" on page 2-18.

Field

The data in this field is used by the create source from retail conversion template process to assign names to your fields in the generated source. You must keep the following items in mind when choosing names for your fields:

For the DB2 Table defined record creation, the system will set the field names to the corresponding database table field name. The RECID##GEN and STORE##GEN fields must be defined as field names in the pre-defined DB2 Table if they are to be used. These will be used by the polling application conver-

sion to help ensure uniqueness of converted retail data.

You must ensure the uniqueness of the field names entered. None of the high-level language compilers, DDS, RPG, or COBOL, permit duplicate field names within the same record format.

For the non-DB2 Table defined record conversion templates, the system uses field names RECID##GEN, STORE##GEN, and DATE###GEN. You should not use these three names. If you use data type 7, see "Special Handling for Data Type 7=Item Code" on page 2-24.

The system allows you to enter field names up to 10 characters long. If you are creating source for RPG the system will truncate these to 6 characters. You must ensure the uniqueness of the first 6 characters in this case.

You must choose names that conform to the naming conventions of the high-level languages for which you will create source. If you create COBOL source, the system attempts to convert any of the extended alphabetic characters (#, &, @, _) to a hyphen. If you are creating DDS source, the system attempts to create a COBOL alias to replace the extended alphabets with hyphens.

Decimal positions

This field must be entered if the AS/400 data type is specified as 3=PACKED. It will be used by source creation to define the field correctly.

Field option and Action field

The field option and Action field parameters are used to set specific fields and make additional omission checks for records base on field values. The field value can be set according to the field option setting. The following is the set of possible combinations.

Field option

'N' No field option

Action field

BLANKS

Field option

'D *value*' Default value

The *value* is stored in the form of the AS/400 data type and length. Currently the max length = 20. So for the AS/400 integer(binary), the first 4 bytes of field value are stored as binary data, the rest is blanks and should be ignored. Character data will be character data followed by blanks.

Action field

BLANKS

Field option

'- *value*' Negative indicator.

The *value* is compared to the converted value of this field. If the comparison is valid (the value is

equal to the converted field) this is a Negative indicator field.

Action field

This is the name of the field in the AS/400 table that is set to a negative value. The field must exist and must be defined previous in the set of field definitions for this record.

Field option

'< *value*' Less than record acceptance check

This is a record acceptance field check. If the field after conversion is less than this field *value*, processing continues for this record. Otherwise, the system omits the record.

Action field

BLANKS

Field option

'> *value*' Greater than record acceptance check

This is a record acceptance field check. If the field after conversion is greater this field *value*, processing continues for this record. Otherwise, the system omits the record.

Action field

BLANKS

Field option

'= *value*' Equals record acceptance check

This is a record acceptance field check. If the field after conversion is equal to this field *value*, processing continues for this record. Otherwise, the system omits the record.

Action field

BLANKS

Data type field size limitations

Most, but not all, combinations of retail data types and AS/400 data types are permissible. The following table illustrates which combinations of the two data types are valid.

Retail Data Type	AS/400 Data Type						
	1	2	3	4	5	6	7
1	X	X	X		X	X	
2	X	X	X	X	X	X	
3	X	X	X		X	X	
4	X	X	X		X	X	
5				X	X	X	
6	X	X	X	X			
7							X

The following field size limitations are in effect. In most cases they are based on the retail data type.

Retail data type Field size limitation

- 1** ASCII. The maximum field size is 256 bytes. Longer fields should be split into multiple pieces.
- 2** Packed. Fields of 1 to 8 bytes may be converted to numeric, but fields over 8 bytes in length must be converted to zoned decimal (EBCDIC-Right).
- 3** Reverse Integer. Fields of 1, 2, and 4 bytes are supported.
- 4** Integer 36xx. Fields of 1, 2, and 4 bytes are supported.
- 5** Flag. All flag fields are 1 byte in length.
- 6** Drop. The maximum drop length is 256 bytes. Longer drop lengths should be specified as multiple fields (fixed-length records only).
Note: If specifying drop fields for a variable length record, each field that you want to drop must be identified.
- 7** Item. See "Special item code handling for data types" on page 2-19.

Retail data type 6 (Drop) considerations

If a retail data type 6 (drop field) is specified in your retail conversion template, this will act as a pad field on a CVTTORTLF command.

If the AS/400 data type is 3 (packed), the retail field will be primed with X'00'. For all other AS/400 data types the retail field will be primed with X'20' (ASCII spaces).

AS/400 data type 6 (Pad) considerations

The CVTFRMRTL command will pad extra space into the output file if an AS/400 data type 6 (Pad output) is encountered. The field will be primed, based on the retail data type, as follows.

Note: The AS/400 system is very particular about the contents of numeric fields. You must ensure that the correct retail data type is specified for Pad output fields in order to avoid MCH1202 system error, decimal data-check errors.

If the retail data type is 1 (ASCII) or 5 (Flag), the AS/400 pad field will be primed with X'40' (EBCDIC spaces).

If the retail data type is 2 (Packed), the AS/400 pad field will be primed with X'00'; except for the right-most byte which will be primed with a X'0F'. This equates to an AS/400 numeric packed decimal field with a value of zero.

If the retail data type is 3 (Reverse integer) or 4 (Integer 36xx), the AS/400 pad field will be primed with X'00' (binary zeros).

Special item code handling for data types

Data type 7 (Item code) is designed to facilitate special item code handling for users of the 4690 General Sales Application (GSA) or the 4690 Store Management Application (SMA). This data type should be used only for the item code field within these applications. If you are not using one of those two store applications, do not use data type 7.

When data type 7 is specified, both the retail data type and the AS/400 data type must be 7. The only lengths that are permissible are 11 and 20. If you are using GSA, you must specify a length of 11. If you are using SMA, you must specify a length of 20. The AS/400 field length will always be 23 for a data type 7 field.

The first byte of the 23 bytes is a flag byte indicating the type and format of data in the remaining 22 bytes. The flag byte is a 1-byte character field.

For GSA, the flag byte has the following meanings:

- 0** UPC/EAN¹ and velocity
- 1** Stock Keeping Unit (SKU)
- 2** Department/Class/Stock (DCS)
- 3** Non-merchandise number

For more information, see the *IBM 4690 General Sales Application: Programming Guide, SC30-3370* and the *IBM 4690 Supermarket Application: Programming Guide, SC30-3373*.

For GSA, the format of the remaining 22 bytes is as follows:

Flag = 0,1,3

The item code number will be represented as a zoned decimal number, right-justified within the 22 bytes, with leading zeros.

Flag = 2

The item code number will be split into three pieces. Each part of the number will be represented as a zoned decimal number, right-justified with leading zeros within a specific allocated section of the 22 bytes. The part of the number representing department will be within the first 6 positions of the 22 bytes. The part of the number representing the class will be in the next 4 positions of the 22 bytes. The part representing stock will be in the last 12 positions of the 22 bytes. If only a department number is present, it will be in its allocated location. The other locations will contain zeros.

For SMA, the flag byte has the following meanings:

- 0** UPC/EAN and velocity
- D** Non-merchandise number
- E** Department/Class/Stock (DCS)

F Stock Keeping Unit (SKU)

For SMA, the format of the remaining 22 bytes is as follows:

Flag = 0,D

The item code number will be represented as a zoned decimal number, right-justified within the 22 bytes, with leading zeros.

Flag = E

The item code number will be split into three pieces. Each part of the number will be represented as a zoned decimal number, right-justified with leading zeros within a specific allocated section of the 22 bytes. The part of the number representing department will be within the first 6 positions of the 22 bytes. The part of the number representing the class will be in the next 4 positions of the 22 bytes. The part representing stock will be in the last 12 positions of the 22 bytes. If only a department number is present, it will be in its allocated location. The other locations will contain zeros.

Flag = F

The item code number (may be alphabetic) will be represented as 20 bytes of EBCDIC character data, left-justified within the field. Following the last significant character, the remainder of the 22 bytes will be padded with EBCDIC spaces.

If you write your own applications on the AS/400 system to load or change files that will be translated using the CVTTORTLF command, and the retail translation template has data type 7 contained within it, you must load the AS/400 data exactly as described above, including the leading flag byte. See "Special Handling for Data Type 7=Item Code" on page 2-24 to understand how to use source generation for this data type and what you have to do in your application code to handle this special data type.

Special Program Considerations:

General Sales Application – Item Code Special Consideration:

If layaways are rung at a point-of-sale terminal for items that have SKU item codes, the 4690 General Sales Application will insert hexadecimal F for all nonkeyed characters in the item code fields of both the Layaway File and the Transaction Summary Log (TLOG). If these files are retrieved back to the AS/400 system and translated using field type 7 (mandatory), the hexadecimal F in the file will be replaced with zeros.

Supermarket Sales Application Special Consideration:

If you are retrieving and attempting to translate the Supermarket Item Movement Totals (EAMIMOVE or EAMIMOVO) file, you will need to define the translation template in one of the following ways.

¹ UPC is the abbreviation for Universal Product Code; EAN is the abbreviation for European Article Number.

If you are positive that there are no item numbers in your file that start with a series of ten 9s, use Record format selector type 2 (Fixed Location), Record selector location 1, and Record selector length 5.

Define a record with the record format selector equal to X'9999999999'. This is your store record.

Define a second record with a Y value specified for the *Catchall record* prompt.

If you are positive that there are no item numbers in your file that end with a series of ten 9s, define your template the same way as the previous method except set the Record selector location to 2.

If you cannot assure yourself of one of the two preceding methods, define a single record format translation template and have your own application code ignore the translated record containing the item number that is all 9s (99999999999).

Chain Sales Special Considerations:

If you are using the IBM 4690 Chain Drug Sales Application, there are special considerations that you must be aware of if you intend to translate the Transaction Summary Log from Retail to the AS/400 system or have your application build files to send to the Price Management Application or the Electronic Mail functions.

The Transaction Summary Log of this application has a different format than the transaction summary logs of the other applications. When defining the retail conversion template, you must indicate Organization 1 (Sequential), Variable length fields as `NO` and String as `YES`.

Special concern must also be given when defining fields within records. For all strings that are not an alert string or preceded by an alert string, define the fields as for any other record format. For those strings that are preceded by an alert string, consider the alert string to be part of the actual data string. Define the 1-byte alert string as the first field of the actual data string. Use the value of the 1-byte of the alert string to be the record format selector for the string that you are defining. Do not use the string type of the actual string. String type 36, Modify UPC Coupon Value Code Table, cannot be translated, due to the large number of fields it contains. It may be passed, if desired, as large blocks of copy data.

The price management file is defined in the manual to be a sequential file using the formats of the price change file. Because of the format of this file, you must define the retail translation template as: Organization 1 (Sequential), Variable length fields as `NO`, and String as `NO`. Do not forget to create the 01 or 02 record as the first record in your file and the 99 record as the last. The Electronic Mail files have the same considerations as the Price Management files.

Steps to Create a Retail Conversion Template

The following instructions will guide you in creating retail conversion templates. The steps necessary are not difficult, however, extreme care must be paid to details in order for the template to convert files from retail format to AS/400 database files and vice versa correctly:

1. The first step is to identify the retail file that will be used to create a retail template. If you are creating a template for a file in an IBM supplied application, the information that you will need will be found in the programmer's guide for that retail controller product.
2. The following information about the retail file must be determined:
 - a) On what type of controller does the file reside (4690, 4684, 3650, or 3680)?
 - b) What type of file is the retail file (keyed, sequential, direct, or random)?
 - c) If b) is sequential, is the file a string file (yes or no)?
 - d) If a) is 3650 or 3680 and b) is keyed, does the file have variable length records (yes or no)?
 - e) Does the file contain multiple record types (yes or no)?
 - f) If c) is yes, determine the format selector type, length and location (first byte is relative position 1, second byte is relative position 2, and so on) in the record.
3. Choose a name to assign to the retail conversion template. The name must be unique among your retail conversion templates. This name will be used whenever you want to do a conversion using this template. Also choose names to be placed into the template as default retail file name and default AS/400 file name. It is recommended that the template name be the same as the name of the file on the retail controller. It is suggested that the two default file names be some variant of that file name.
4. Access the Work with Retail Conversion menu and select option 3 (Work with retail conversion templates). Use F6 to add a retail template. Use the data gathered in step 2 to respond to the prompts.

Note: There are two displays that must be responded to when adding a template. Paging down (ROLLUP) will access the second display after responding to the first. Press the Enter key after responding to the second display.
5. Gather information about the record formats within the retail file. If there is only a single record format, decide on a name to use for this format. You may use the template name as a record name. If there are multiple record formats, determine the record format selector for each record type in the file. Determine if you need a catchall record. Assign a name to each of these record

format selectors and to the catchall. Decide which of these record types you want the conversion to process.

Note: Record format names must be unique within a given retail conversion template.

6. From the Work with Retail Conversion Templates display, select option 8 (Work with records), for the retail conversion template that you are creating.
7. You will need to use F6 (Add record) as many times as there are records types within your file.
Note: Each record format in the file MUST be described whether you intend to convert it or not. If you do not intend to convert it specify Omit record yes (Y) when adding the record.
8. For each record added in step 7, determine the retail record layout at a field level. You will need to know the type of data in the retail field (packed decimal, ASCII, or flag, for example) and the length of the retail field in bytes (not in number of digits).
9. For each field listed in step 8, determine what type of conversion you want to make into an AS/400 file. This includes the possibility of dropping the data from the conversion and also copying the retail data as is (no conversion) into the AS/400 database file. You may also choose to leave extra space between fields after they have been converted. You will need to assign a name to each of the fields. This name will be used by the Source Generation Facility to generate source code for you to use. It is your responsibility to insure that the names you select conform to the naming conventions of the language for which you will generate source.
10. From the Work with Records in Retail Template display, select option 8 (Work with fields) for each of the records within your template.
11. Use F6 (Add field) for each field identified in steps 8 and 9. The fields should be entered in order as they appear in the retail file. It is recommended that you assign sequence numbers in multiples of 10 to help you in inserting more fields, if necessary, at a later time. Sequence numbers may be reset at any time using F20 (Renumber) on the Work with Fields in Template Record display.
12. When you have completed entering all fields for all records within the template, use F3 one or two times until you reach the *Exit Work with Templates* confirmation prompt. Choose option 1 (Exit and save changes). It is recommended that you respond Yes to the *Print Template* prompt to enable you to more easily review your work for accuracy. The printed output will be placed on your job's output queue.
13. Your retail conversion template has now been created. You may proceed to create other templates. Before using your template in a production environment, it is recommended that you test it thoroughly. This will probably entail generating source statements, receiving files from your retail controller, converting the files, and checking the results.

Steps to Create a Retail Database Collection Conversion Template

The following instructions will guide you in creating retail conversion templates for converting to a DB2 Collection.

1. The first step is to identify the retail file that will be used to create a retail template. If you are creating a template for a file in an IBM supplied application, the information that you will need will be found in the programmer's guide for that retail controller product.
2. The following information about the retail file must be determined:
 - a) What type of file is the retail file (sequential, direct, or random)?
 - b) If a) is sequential, is the file a string file (yes or no)?
 - c) Does the file contain multiple record types (yes or no)
 - d) If b) is yes, determine the format selector type, length and location (first byte is relative position 1, second byte is relative position 2, and so on) in the record.
3. Gather information about the record formats within the retail file. If there is only a single record format, decide on a name to use for this format. You may use the template name as a record name. If there are multiple record formats, determine the record format selector for each record type in the file. Decide which of these record types you want the conversion to process.

Note: Record format names must be unique within a given retail conversion template. The DB2 Table name does not have to be unique. So multiple records may be created for each record format.

4. Create the Relational Database collection using the information gathered in the previous step. This can be accomplished using SQL or DDS. Each field defined in the database should correspond to the Retail file formats, and should be convertible by the polling application.

Note: The DB2 Collection must exist before the corresponding template can be created. It is also not necessary to convert/insert all the retail data into a AS/400 database table. You may choose to eliminate some fields when creating you database tables. If this is done, make sure the retail fields are added to the field template field definitions as drop, so the polling application does not process this data incorrectly.

Note: All relational database collection keyed fields must be in the same order and offset of the constraining DB2 Table for the polling application to process the relational keyed field correctly. For example: If a "Header" file is created with a key of StoreNbr, RegisterNbr and TransNbr, and a file is created that is constrained on "Header",

then it must have the StoreNbr, RegisterNbr, and TransNbr in the same order.

5. Choose a name to assign to the retail conversion template. The name must be unique among your retail conversion templates. This name will be used whenever you want to do a conversion using this template. Also choose names to be placed into the template as default AS/400 file name. It is recommended that the template name be the same as the created DB2 Collection.

6. Access the Work with Retail Conversion menu and select option 3 (Work with retail conversion templates). Use F6 to add a retail template. Use the data gathered in step 2 to respond to the prompts.

Note: There are two displays that must be responded to when adding a template. Paging down (ROLLUP) will access the second display after responding to the first. Press the Enter key after responding to the second display.

7. From the Work with Retail Conversion Templates display, select option 8 (Work with records), for the retail conversion template that you are creating.

8. You will need to use F6 (Add record) at least as many times as there are records types within your file.

Note: Each record format in the file MUST be described whether you intend to convert it or not. If you do not intend to convert it specify Omit record yes (Y) when adding the record.

9. For each record added in step 7, determine the retail record layout at a field level. You will need to know the type of data in the retail field (packed decimal, ASCII, or flag, for example) and the length of the retail field in bytes (not in number of digits).

10. For each field listed in step 8, determine what corresponding field is defined in the database table. This includes the possibility of dropping the data from the conversion and also copying the retail data as is (no conversion) into the AS/400 database file. You will need to assign a name to each of the fields that you add that is not defined in the DB2 Table.

11. From the Work with Records in Retail Template display, select option 8 (Work with fields) for each of the records within your template. This will cause the template support to retrieve the Field descriptions out of the DB2 Table defined on the record entry.

Note: For DDS described tables, the record format name in the DDS MUST be the same name as the physical file name created, and the DB2 Table name defined on the record entry.

12. Use F6 (Add field) or option 2 (Change field) for each fields identified in steps 8 and 9. The fields should be entered in order as they appear in the retail file. It is recommended that you assign sequence numbers in multiples of 10 to help you in inserting more fields, if

necessary, at a later time. Sequence numbers may be reset at any time using F20 (Re-number) on the Work with Fields in Template Record display.

Note:

It is possible to define the DB2 Table fields so that they do not have the same order as the retail data. For these cases, the sequence number should be changed to order the data in the retail file defined order.

This is not valid for zoned flag data that is to be converted using the string template conversion. All the string flags must be in the 4690 represented order.

13. When you have completed entering all fields for all records within the template, use F3 one or two times until you reach the *Exit Work with Templates* confirmation prompt. Choose option 1 (Exit and save changes). It is recommended that you respond Yes to the *Print Template* prompt to enable you to more easily review your work for accuracy. The printed output will be placed on your job's output queue.

14. Your retail conversion DB2 Collection template has now been created. Before using your template in a production environment, it is recommended that you test it thoroughly. This will probably entail generating source statements, receiving files from your retail controller, converting the files, and checking the results.

Retail File Conversion Considerations

RAIS has 2 conversion mechanisms that work with the template definitions to convert retail application data.

The conversion commands in the Point of Sale/2000 option (option 1) are used to convert retail application flat file data to AS/400 data or AS/400 format data to retail application data. This file conversion system supports a maximum of 100 record types in a template and a maximum of 214 fields in a record. The maximum retail file record length supported is 256 bytes for the 3651/3684 and 768 bytes for the 4690 or 4684. For 4690 string files, this limitation applies to the length of each string, not record. The maximum record length for an AS/400 file is 1536 bytes. Each record type in a multiple record format file must be identifiable by a record code of up to 5 bytes in a fixed location, field number in the retail record, or by relative record number (up to 5 digits) of the retail record. Record formats not explicitly defined can be translated with a catchall record format.

The polling application (option 2) allows for the conversion of retail application data to an AS/400 format data. For DB2 Collection referenced templates, this application will insert the converted data into the relational data base collection or other pre-defined Physical files. See Chapter 5, "RAIS Retail Polling Application" on page 5-1

Data Conversion Routine

See the *Retail Communications Programming* book for data conversion information.

Retail Conversion Templates

You can display and select from a list of available conversion templates. The template name can also be specified for batch conversion jobs. You are given the capability to inter-actively change or add unique file conversion templates.

User Conversion Requirements

If you do not want to use the retail conversion routines supplied with the Retail Application Interface Suite, you can write your own data conversion programs.

To do this, you must be familiar with the retail file formats and data representation of the retail files to be converted. You will find program QRFXTL (provided as part of AS/400* program) extremely valuable. This program is described in the *Retail Communications Programming* book.

If you want to use the retail file conversion routines supplied with the Retail Application Interface Suite, use this chapter as your guide.

HCP Host Conversion Services: File transmissions to or from the host can be PC binary format, programmable store system (PSS) format, new host format, or PC print format. HCP provides varying degrees of file conversion services for these different transmitted file formats.

The PC binary format transmits exact disk sector images from the store controller with no conversion at all. This method uses the least resources on the retail controller, but requires coding on the host system to perform data conversion. This coding has been provided for you through the file conversion system of the utility, which is the easiest method to use on the host system.

The PSS data formats are designed to provide compatibility with previous IBM PSS point-of-sale systems, so that the host system can support a mixed network of current and prior systems.

The new host format is designed to provide a compromise between host ease of use and transmission time.

The PC printer file request works only for character data files (print images). The PC ASCII file must be less than 255 characters and end with a carriage return or line feed. The host ECBDIC file is converted from ASCII to ECBDIC and has a 1-byte (binary) header with the length of the following data record.

Point-of-Sale Template Records and Fields

Incoming files with multiple record formats are accepted. Up to 100 different record formats can be specified for each template. Each record type must be identifiable by either a record code (up to 5 digits) in a fixed location, a field number within the record, or by relative record number (up to 5 digits). Additionally, a **catchall record** format may be described for all records not meeting the requirements of an explicitly defined record format.

Each point-of-sale input record must be described in terms of specific point-of-sale data fields. Each field must have a maximum input length, a point-of-sale input data type, and an output data type. In addition, a field name and number of decimals should be provided if the source statement creation option is to be used.

Point-of-Sale Data Fields Conversion

In building the point-of-sale conversion template for non-DB2 Collection referenced support, you must indicate both the incoming data type and outgoing data type. For DB2 Collection referenced templates, the database files must exist before the creation of the Template, and the system will indicate the AS/400 data types using the physical file description as a reference. The user will have to update the Retail data type for these cases. For information on conversion data types and building point-of-sale conversion templates, see "Work with Retail Conversion Templates" on page 2-7.

Point-of-Sale Conversion High-Level Source Creation

The high-level source creation facility, accessed using the Create Retail Source (CRTRLSRC) command or the Work with Retail Templates display, is provided to allow you an easy method to process data in files that have been converted from retail format into AS/400 database files. Three kinds of high-level source can be obtained: data description specifications (DDS), RPG/400, and COBOL/400.

DDS

You can generate DDS from a retail conversion template. If the template is for a single record format file, the DDS can be used as direct input to a Create Physical File (CRTPF) command for creating an externally described file on the AS/400 system to receive your converted data. If the template is for a multiple record format file, specifications for each format are generated into the source member with comments separating each format. COBOL ALIAS names are generated for all fields that contain the special characters #, \$, @, and _. No attempt is made to avoid the use of COBOL/400 reserved words as field names; this is your responsibility during template creation. No attempt is made to ensure that field names are 6 characters or less for use in RPG/400; this is your responsibility during template creation.

RPG/400

You can generate RPG/400 input and output specifications from a retail conversion template. The specifications are generated at the field level only. No record level Input or Output specifications are generated. If the template you are creating source from has multiple record formats, the formats are separated in the source file with comments lines. Field names in the template that are more than 6 characters long are truncated to 6 characters. It is your responsibility to ensure that name truncation does not cause duplicate field names if these specifications are used in your RPG/400 programs.

COBOL/400

You can generate COBOL/400 data division specifications from a retail conversion template. Each record format in the retail template causes an 01 level entry named RRRRRRRRRR-of- TTTTTTTTTT, where RRRRRRRRRR is the name assigned to the record in the conversion template and TTTTTTTTTT is the name of the conversion template. This is followed by 05 level entries for each field in the template. No attempt is made to ensure against the use of COBOL/400 reserved words as field names; this is your responsibility during template creation. No FD or SELECT statements are generated.

Special Handling for Data Type 7=Item Code

If your template contains special data type 7=Item code, the source will generate fixed output in the appropriate location within the record. The field name that you assign in the template will **not** be used. Output field names are fixed.

DDS

The source will place two fields into your DDS representing the item code. The first field will be named ITEM##FLAG (with appropriate COBOL ALIAS) and will be a character field 1 byte in length. The second field will be named ITEM##CODE (with appropriate COBOL ALIAS) and will be a character field 22 bytes in length.

RPG/400

The following source code will be generated for RPG. This example assumes that the item code will start in byte 2 of the AS/400 file.

```
I          2  2 IT##FL
I          3 22 IT##SK
I          3 24 IT##CD
I          3  8 IT##DP
I          9 12 IT##CL
I         13 24 IT##ST

O          IT##FL      2
O
O  INDICATORS REQUIRED BASED ON FIELD TYPE
O
O          IT##SK      22
O          IT##CD      24
O          IT##DP       8
O          IT##CL      12
O          IT##ST      24
O
O  END INDICATOR REQUIREMENT
O
```

Because the same positions are multiply defined in the output specification, RPG/400 users will have to supply indicators for all IT## fields (except IT##FL) based on the type of item code being output.

COBOL/400

The following source code will be generated for COBOL:

```
5  ITEM--FLAG      PICTURE X( 1).
5  ITEM--SKU       PICTURE X( 22).
5  ITEM--DCS       REDEFINES      ITEM--SKU.
6  ITEM--DEPT     PICTURE S9( 6).
6  ITEM--CLAS     PICTURE S9( 4).
6  ITEM--STOK     PICTURE S9(12).
```

See “Fields within Retail Conversion Template Records Descriptions” on page 2-15 for more information on the special data type 7=Item code as well as all other data types.

Using Generated High-Level Source Statements:

For retail conversion templates that have a single record format, the recommended approach is to create DDS source. Use this generated source to create a physical file. Then compile this file into your high level-language programs as you would any other externally described file. For this approach you should use the externally created file you have created as the output from the CVTFMRRLF command and the input to the CVTTORRLF command. You can change the DDS prior to physical file creation to add one or more key fields.

If you have more than one record format and you are omitting all but one format from the conversion, the recommended approach is to create DDS source. Change the source generated to remove all specifications that pertain to all record formats other than the one being converted. When only the format being converted is left, create a physical file from the DDS. Use this file as an externally described file in your high-level language programs. For this approach use the externally created file as the output from the CVTFMRRLF command and the input to the CVTTORRLF command.

If you have more than one record format to be converted, you have a choice in processing. You must decide whether you want to use externally described files for the converted data. The recommended approach is to use program-described files for the converted data.

Externally Described Files with Multiple Formats

1. Check and change the template to ensure that the record code of each record will be converted to a data type that you can check using either a Copy File (CPYF) command or a high-level language program.

This is not necessary if your high-level language programs are written in either COBOL/400 or RPG/400. The conversion process always places the record code found in the retail file in the first bytes of the converted output in retail image (you can use hexadecimal to check it).

2. Create DDS source from the retail conversion template.

3. Using the source entry utility (SEU), separate each of the DDS formats created into separate source members.
4. Create a physical file from each of the separated members.
5. Additionally, create a nonfield level physical file using the maximum record length displayed on the top of the Work with Records in Retail Template display.

This is to be used as the value for the RCDLEN parameter on the CRTPF command for the retail conversion template in question. If you want to process the various record formats sequentially, you can create a logical file spanning all of the records you intend to process.

6. Convert the incoming retail file into the nonfield level file.
7. Using either the Copy File (CPYF) command multiple times or a high-level language program, read each record in the nonfield level file.
8. Write this record, based on the hex record code in the first bytes of each record or the translated record code, to the appropriate physical file or to the appropriate record format of a logical file spanning multiple physical files.

Internally Described Files with Multiple Formats

1. Create high-level source for the language you are using; either RPG/400 or COBOL/400. Include these specifications in your application program. For RPG/400, you cannot define the record code in the first bytes on record-level I specs.
2. Code the input file in your application programs as a program-described file.
3. Create a nonfield level file using the maximum record length as displayed on the top of the Work with Record in Retail Template for the retail conversion template as the value for RCDLEN on the CRTPF command.
4. Convert your incoming retail data into the nonfield level file. Use this file as input to your application program.

Create Retail Source (CRTRTL SRC) Command

The Create Retail Source (CRTRTL SRC) command creates high-level language source to be used in processing converted retail files. The created source can be used directly or copied into RPG/400 or COBOL/400 programs.

```

Job: B,I Pgm: B,I REXX: B,I Exec
Threadsaf: No

CRTRTL SRCRTLPL( retail-template-name )
                                DDS
                                SRCTYPE(   RPG   )
                                CBL
                                (P)
SRCMBR(   RTLTPL
member-name   ) SRCFILE(   LIBL/
                                CURLIB/
library-name/   SRCTYPE
source-type
source-file-name
)
MBROPT(   REPLACE
NONE   )

```

Note:

^P All parameters preceding this point can be specified positionally.

RTLTPL Parameter

This parameter specifies the name of the retail conversion template from which you want to create AS/400 source statements.

SRCTYPE Parameter

This parameter specifies the type of source code to be generated.

*DDS

DDS source specifications are to be created. If the retail conversion template has only a single record format, this source can be used directly as input to a CRTPF command.

*RPG

RPG/400 source input/output specifications are to be created. Field level I and O specs are generated. Comments are generated to indicate the start of each record format. F specs are not generated.

*CBL

COBOL/400 source data division specifications are to be created. A data division data structure is generated for each record defined in the retail conversion template.

SRCMBR Parameter

This parameter specifies the name by which the physical file member to contain the source is known. The source member need not exist prior to running the command.

***RTLTP**

The name of the physical file member to contain the source is the same as the name of the template.

member-name

Specifies the name of the member to contain the source.

SRCFILE Parameter

This parameter specifies the qualified name of the source physical file that is to contain the generated source (library name/file name). The source file must exist prior to running of this command.

The possible values area:

***LIBL**

The library list is used to locate the file name.

***CURLIB**

The current library for the job is used to locate the file name. If no current library entry exists in the library list, QGPL is used.

library-name

Specifies the library where the file is located.

***SRCTYPE**

Specifies that the value specified for SRCTYPE parameter determines the source file to be used.

source-type

Specify one of the following source types:

***DDS** File QDDSSRC is used.

***RPG** File QRPGRSRC is used.

***CBL** File QLBSLRC is used.

source-file-name

Specifies the name of the source physical file in which to place the generated source.

MBROPT Parameter

This parameter specifies, for existing source members, whether the data currently in those members should be replaced.

***REPLACE**

Specifies to replace existing source with newly generated source.

***NONE**

Specifies to not replace existing source with newly generated source.

If the physical file member specified in the SRCMBR parameter does not exist in the source file specified in the SRCFILE parameter, a new member will be added to the source file.

The following error messages can be generated from this command:

POS9611 POS9612 POS9613 POS9614

Print Retail Conversion Template (PRRTLTP) Command

The Print Retail Conversion Template (PRRTLTP) command creates a formatted list of the contents of a retail conversion template.

```

                                Job: B,I Pgm: B,I REXX: B,I Exec
                                Threadsaf: No
PRRTLTP( retail-template-name ) (P)
                                ALL
                                RCDfmt( record-format-name )

```

Note:
P All parameters preceding this point can be specified positionally.

RTLTP Parameter

This parameter specifies the name of the retail conversion template which you want to print.

RCDfmt Parameter

This parameter specifies which record definition within a retail conversion template is to be printed.

***ALL**

Print all records found.

record-format-name

Prints only the single record specified.

The following error messages can be generated from this command:

POS9601 POS9602

Work with Retail Templates (WRKRTLTP) Command

The Work with Retail Templates (WRKRTLTP) command permits you to maintain the retail conversion templates on your system. Templates can be added, changed, copied, renamed, or deleted. Similar functions are available within the template for template records and fields.


```

                ALL
WRKRTLTPIRRTLPL(  retail-template-name  (P) )
                generic -template-name

```

Note:

^P All parameters preceding this point can be specified positionally.

RTLTPL Parameter

This parameter specifies the name of the retail conversion template you want to work with.

***ALL**

All retail conversion templates are to be displayed.

retail-template-name

Specifies the name of the retail conversion template you want to maintain.

generic*-template-name

Specifies the generic retail template name for the retail conversion templates to be maintained. To

specify a generic name, add an asterisk after the last character in the generic name (ABC*, for example). If an * is not included with the name, the system assumes that the name is a complete retail conversion template name.

Chapter 3. Retail File Application Programming Interfaces

This chapter gives you the information needed to work with the base option retail communications application programming interfaces. RAIS has 2 APIs which can be used. The first is the ADCS Emulation API which is used strictly, with the exception of the acquire function, in LU0 environments. The second is the RAIS API which can be used in an LU0, TCP/IP, or Local environment. A detailed description is provided for each API.

ADCS Emulation Application Program Interface (API)

General-Use Programming Interface

While retail communications commands provide high-level access to the ADCS functions, there are some interactive applications that cannot be performed without additional programming. One of these applications is real time point-of-sale file maintenance from an AS/400 application. An API is provided for the ADCS emulation program to handle these cases.

This ADCS emulation API gives user-written programs access to the following ADCS functions:

ACQ

For LU0 location: allows the user to start (acquire) a communications session with a retail controller. The ACQ operation must be performed before any other API function. For a TCP/IP or LU6.2 location: allows the user to check the connection to this location.

ADD

For LU0 location: add or replace a retail keyed record.
For TCP/IP location: not supported.

DEL

For LU0 location: delete a retail keyed record. For TCP/IP location: not supported.

READ

For LU0 location: read a retail keyed record. For TCP/IP location: not supported.

REL

For LU0 location: allows the user to end (release) a communications session with a retail controller. This should be the last API function performed. For TCP/IP location: no operation performed.

The file level commands are not supported by the API routine. You can perform these functions in your program by coding the appropriate instructions or by calling the CL commands provided with this licensed program.

The API can be accessed through a call routine from AS/400 batch or interactive programs.

COBOL/400 and RPG/400 API Coding Examples:

The API call routine requires a user-described parameter list.

The following is a COBOL/400 example of the call and parameter list for the ADCS emulation API.

```

1 API-PARM-LIST.
  3 FUNCTION          PIC X(4).
  3 FILE-NAME         PIC X(6).
  3 LOCATION          PIC X(8)
  3 RECORD-LENGTH    PIC S9(3) USE IS COMP-3.
  3 RECORD-DATA      PIC X(256).
  3 END-OF-DATA      PIC X.
  3 FAIL-TYPE        PIC X.
  3 RETURN-CODE      PIC X(4).
  3 RETURN-DATA-PORT PIC X(8).

CALL 'QRTLAPI' USING FUNCTION FILE-NAME LOCATION
                    RECORD-LENGTH RECORD-DATA
                    END-OF-DATA FAIL-TYPE RETURN-CODE RETURN-DATA-PORT.

```

Figure 3-1. COBOL/400 API Coding Example

The following is an RPG/400 example of the call and parameter list for the ADCS emulation API.

```

C          CALL      'QRTLAPI'  APILST
C
C  APILST  PLIST
C          PARM          APIFN      4
C          PARM          APIRTL     6
C          PARM          RMTLOC     8
C          PARM          RCDLEN     3
C          PARM          RCDDTA    256
C          PARM          EOD        1
C          PARM          FALTYP     1
C          PARM          RTCD       4
C          PARM          RTDTPT     8

```

Figure 3-2. RPG/400 API Coding Example

The parameters listed in Figure 3-2 should contain the following data:

FUNCTION (APIFN)

APIFN is the API function to be performed. These values should be left-justified within the field. The valid functions available are the following:

ACQ

For LU0 location: allows the user to start (acquire) a communications session with a retail controller. The ACQ operation must be performed before any other API function. For a TCP/IP or LU6.2 location: allows the user to check the connection to this location.

ADD

For LU0 location: allows the user to add or replace a retail record to a retail file. For TCP/IP or LU6.2 location: not supported.

DEL

For LU0 location: deletes a keyed retail record from a retail file. For TCP/IP or LU6.2 location: not supported.

READ

For LU0 location: reads a keyed record from a retail file. For TCP/IP or LU6.2 location: not supported.

REL

For LU0 location: allows the user to end (release) a communications session with a retail controller. This should be the last API function performed. For TCP/IP or LU6.2 location: no operation performed.

FILE-NAME (APIRTL)

APIRTL is the 6-character name of the keyed file on the retail controller on which the requested API-FUNCTION will be performed.

See RTLFILE parameter of "Send Retail Record (SNDRTRLRCD) Command" on page 4-17 for an explanation of this value.

LOCATION (RMTLOC)

RMTLOC is the remote location name specified on the AS/400 device description.

RECORD-LENGTH (RCDLEN)

RCDLEN is the record length in bytes of the AS/400 data that you want to send. If the AS/400 data is in host format, be sure to include the carriage-return and line-feed characters (X'0D0A') in the length calculation. Valid values are from 0 to 256. HCP does not support record lengths greater than 256 bytes when working with record level commands.

RECORD-DATA (RCDDTA)

RCDDTA is a 256 byte buffer that should contain the actual record for the keyed file update on the retail controller. Following is an example of what the record would look like for a ADD keyed record for a General Sales Application Item Master file.

This example is in host format. See the *IBM 4680 Store System: Programming Guide* for information on host format records.

```
123456789, , , ,999,3,19999, ,item description, ,##
```

Note that this record is in HOST format and fields are delimited by commas. All records are ended with X'0D0A' for carriage return and line feed. The X'0D0A' is shown as a ## in the data example because they are unprintable hex characters. For a DELETE or READ only the key would be required, 00123456789. The record length must include 2 bytes for the X'0D0A' (Carriage return /Line feed), so the record length for this example is 53. This example maps the GSA Item Master File in the *IBM 4680 General Sales Application (GSA): Programming Guide*.

The first field is the key of item number, and remaining fields are the indicator bytes, department number, description, sale quantity and price. (See the *IBM 4680 GSA Programming Guide* for file layouts or the *4680*

Store System: Programming Guide, Managing your Files for more information on files.)

END-OF-DATA (EOD)

This flag indicates the end of a series of record updates. Y indicates the end, N means additional records exist. For example, if the user would like to do five price changes, for the first four calls place an N in this field. On the last record place a Y. The N option is provided to increase performance when doing multiple add, replace, or delete operations. The Y option should be used when precise error recovery is desired.

FAIL-TYPE (FALTYP)

This value allows for different types of updates with the retail controller program.

The following codes are allowed:

- R** Indicates an ADD record with the REPLACE option will be used. When using this option, the retail controller sends back error messages for records that do not exist.
- A** Indicates an ADD record with no replace. HCP checks to see if the record exists on the retail controller. If it does, the ADD is not run and an error message is sent back to the AS/400 system.
- B** Indicates that an ADD/REPLACE command be used and error messages are not returned to the AS/400 system. This means that the record is added if it does not exist, or replaced if it does, (HCP does not indicate which one it does).
- Y** For a Delete command, this indicates that if the record does not exist on the retail controller, an error message is returned.
- N** For a Delete command, this indicates that if the record does not exist on the retail controller, no error message is returned.

RETURN-CODE (RTCD)

RTCD is a 4-character number returned from the ADCS emulation program. Zero indicates that the update was successful. Nonzero indicates a failure occurred. Errors are logged in the user's job log.

RETURN-DATA-PORT (RTDTPT):

RTDTPT is an 8-character number returned from the ADCS emulation program. This could contain a major or minor return code, an errno value(see "AS/400 UNIX Error Codes" on page E-1), the file status, or the name of a retail file name. This information can be useful during problem analysis. This field is also used by the TCP/IP ACQ function and an input parameter for the TCP remote port. When specifying the input port value this should be a numeric data in the range of 1-65535 and should be delimited by a space or 'x00'. If the port is specified incorrectly, the default RCO port(20000) is used.

API Return Codes and Examples: The following return codes are possible:

RAIS Communications Application Program Interface (API)

The retail communications commands provide high-level access to the RAIS API functions. However, there are certain instances where additional programming is sometimes necessary. Using the following APIs can assist you in accomplishing this tasks and can be generically designed for a LU0 or TCP/IP environment. The RAIS LU0 support uses the Advanced Data Communication for Stores (ADCS) emulation when communicating with the retail controller. The RAIS TCP/IP support uses Retail Connectivity Option commands when communicating with the retail controller Retail Connectivity Option server.

Note: This is an ILE interface. For more information on the ILE programming, see the *ILE C for AS/400 Programmer's Guide*, *ILE RPG for AS/400 Programmer's Guide*, or *ILE COBOL for AS/400 Programmer's Guide*.

Using the RAIS API

The RAIS API consists of the `qzcfpcmn` service program. When you create your program you will need to bind the `qzcfpcmn` service program by specifying it on the `CRTPGM` command.

```
crtpgm pgm(program library/name)
module(module library/name)
bndsrvpgm(qrais/qzcfpcmn)
```

Linking the service program enables the application to use the retail file APIs to access files on the retail controllers. It includes a set of file functions and a set of environment variables for tailoring the API for a specific application.

RAIS API file functions

The API file functions are structured similar to the AS/400 UNIX file commands. To work with a file on the retail controller an application would first have to open the file description using the `popen()` function. Included in the parameters of the `popen()` is the location of the retail controller to be accessed. Again, this location could be for one of the protocols supported. After a file open is complete and successful, the application can then read(`pread()`) from the file, write(`pwrite()`) to the file, and seek(`pseek()`) to a new location, just as it would a local file. After processing the file description can be closed(`pclose()`). The API also includes function to create(`pcreat()`), delete(`punlink()`), retrieve file information(`pstat()`), and check for file existence(`paccess()`). For keyed file manipulation, after an open of a keyed file, the API has function to read a record(`preadk()`), write a record(`pwritek()`), update a record(`pupdatek()`) and delete a record(`pdeletek()`). Finally, non-file related function includes a start user program function(`psup()`) and a verify connection function(`pping()`).

RAIS API Environment Variables

The Environment variables may be used to tailor the service program for a specific application. The following Environment Variables exist:

- QZCFPROTO - Protocol for location APIs.
- QZCFAPPL - Retail Application for HCP name conversion
- QZCFCNV - Session/Conversation Keep active value
- QZCFWAIT - Server response wait time value
- QZCFATYPE - Retail application file name type

The Environment variables can be set using the Change or Add Environment Variable system API `putenv()`, the Add Environment Variable(`ADDENVVAR`) `CL` command, or the Change Environment Variable (`CHGENVVAR`)`CL` command. The environment variables can be retrieved using the Get Value of Environment Variable system API `getenv()`.

QZCFPROTO: This environment variable is the global location protocol. Its default is `ZCFSEARCH`. Setting this will cause all the Location functions to use this setting as the location protocol. If `ZCFSEARCH` is set, the location functions will calculate the protocol when invoked.

(char[2]) Possible values are:

- ZCFSEARCH (-1) default
- ZCFSNA (0)
- ZCFIP (2)
- ZCFLOCAL (3)

Example

```
#include <stdlib.h>
#include "QRAIS/h/qzcfpcmn"
char EnvironmentVariable[3 ];
int rc;
sprintf(EnvironmentVariable,"%s%i",
        "QZCFPROTO=",ZCFIP);
rc = putenv(EnvironmentVariable);
```

QZCFAPPL: This environment variable is the retail application variable. Its default is `ZCFGSA` or "eal". This variable is used when converting the HCP logical file names for use by the TCP RCO connections.

(char[3]) Possible values are:

- ZCFGSA ("eal") default
- ZCFSA ("eam")
- ZCFCSA ("egh")

Any user defined 3 character application prefix.

Example

```
#include <stdlib.h>
#include "QRAIS/h/qzcfpcmn"
char EnvironmentVariable[3 ];
int rc;
sprintf(EnvironmentVariable,"%s%s",
        "QZCFAPPL=",ZCFGSA);
rc = putenv(EnvironmentVariable);
```

QZCFCNV: This environment variable is the retail session/conversation variable. Its default is ZCFDROP. This variable is used to control whether the session/conversation will be kept up as the application is active, or will be dropped and reestablished when needed. If ZCFKEEP is specified, the session/conversation will be dropped when the application ends.

For LU0 sessions, This flag is specified when the HCP session does need to say up between reads. This essentially locks the session from other system users until the file is closed.

(char[1]) Possible values are:

ZCFDROP ("0") default
ZCFKEEP ("1")

Example

```
#include <stdlib.h>
#include "QRAIS/h/qzcfpcmn"
char EnvironmentVariable[3 ];
int rc;
sprintf(EnvironmentVariable,"%s%s",
        "QZCFCNV=",ZCFKEEP);
rc = putenv(EnvironmentVariable);
```

QZCFWAIT: This environment variable is the RCO response timer. Its default is 30 seconds. This variable is used to specify the time the application should wait for a response from the RCO server. A value of ZCFINDEF will cause the application to wait indefinitely.

(char[4]) Possible values are:

30 seconds default
ZCFINDEF (-1)
0 - 3600 seconds

Example

```
#include <stdlib.h>
#include "QRAIS/h/qzcfpcmn"
char EnvironmentVariable[3 ];
int rc;
sprintf(EnvironmentVariable,"%s%i",
        "QZCFWAIT=",ZCFINDEF);
rc = putenv(EnvironmentVariable);
```

QZCFFTYPE: This environment variable is the retail application file name type. The file name can be an HCP-formatted name, 4680/4690 Logical name, or a 4680/4690 stream file name.

(char[1]) Possible values are:

ZCFHCP 0
ZCFLGLF 1
ZCFSTMF 2

Example

```
#include <stdlib.h>
#include "QRAIS/h/qzcfpcmn"
char EnvironmentVariable[3 ];
int rc;
sprintf(EnvironmentVariable,"%s%i",
        "QZCFFTYPE=",ZCFLGLF);
rc = putenv(EnvironmentVariable);
```

Note: The environment variables are set in the file transfer commands discussed in "Working with Retail Communications" on page 2-4. When using these commands along with an application program written to this API, you may need to reset the environment variables appropriately.

RAIS API location function

The location APIs; popen(), pcreat(), punlink(), psup(), pping(), pstat(), and paccess(), all use the location parameter to specify the configuration used to access the remote location. These APIs will automatically determine the protocol to use for the function call. If this processing is not desired, the QZCFPROTO environment variable can be set. This will force all location calls after the setting to use the specified protocol. The calculating nature of the location functions can be reset by using the ZCFSEARCH value for QZCFPROTO. An additional api that can be used to determine the protocol of some location is the pgetprotocol() function.

RAIS API file naming

A path/file name is passed on the following functions; popen(), pcreat(), punlink(), pstat() and paccess(). This is the name of the file to access on the retail controller. This file name can be one of the following:

HCP 6 character logical name
Application Logical name (defined on the retail controller)
User Logical name (defined on the retail controller)
path/filename.ext

The RAIS API will make any necessary conversions, depending on the location protocol and retail application, to access the file on the retail controller. For example, a HCP file name of "/TRANS" will be converted to

c:\adx_idt1\EALTRANS.DAT when using the TCP protocol and GSA retail application.

For protocol compatibility reasons it is best not to use the user or application logical names. The pstat() and paccess() functions can not resolve to the appropriate directory for HCP when using these logical Names.

Using ILE COBOL/RPG Program to Call APIs

The exported names for the api calls must be used. These must be lower cased.

- qzcfopen
- qzcfcreat
- qzcfread
- qzcfwrite
- qzcfclose
- qzcfstat
- qzcfaccess
- qzcfunlink

- qzcfreadk
- qzcfwritek
- qzcfdeletek
- qzcfupdatek
- qzcfsup
- qzcfping
- qzcfgetprotocol

COBOL modules MUST be created using the *NOMONPRC option since our EXPORTS are lower case.

QZCFERRNO data export is provided to return the api function errno values.

O_FLAG constants are not provided for COBOL or RPG.

STAT structure is not provided

Path names and Location names MUST be null delimited.

See the "RAIS API Example" on page B-1 for an ILE COBOL example using the API.

RAIS Communications API Reference

RAIS Communications API header file: The communications api header file that is used for all of the functions is QZCFPCMN. This should be include in any module that uses the RAIS Communications API.

```

/
/
/ Retail API file descriptor
/
/
typedef int PCMND;

/
/ The following constants are used for determining the remote
/ location protocol using pgetprotocol().
/ These values will be returned by the function.
/ The following constants also can be used for setting the
/ protocol type(QZCFPROTO) Environment Variable.
/
#define ZCFSEARCH -1
#define ZCFNOTFND -2
#define ZCFERROR -3
#define ZCFSSNA
#define ZCFIP 2
#define ZCFLOCAL 3

/
/ The following constants are used for setting the Retail
/ control file name type(QZCFATYPE) Environment Variable.
/ The file name type can be an HCP formatted name,
/ a 469 Logical name, or a 469 stream file
/ name(C:/directory/file.ext).
/
#define ZCFHCP
#define ZCFGLGF 1
#define ZCFSTMF 2

/
/ The following constants are used for setting the retail
/ application type(QZCFAPPL) Environment Variable.
/
#define ZCFGSA "eal"
#define ZCFSA "eam"
#define ZCFCSA "egh"

/
/ The following constants are used for setting the retail
/ conversation/session control (QZCFENV) Environment Variable.
/
#define ZCFDROP " "
#define ZCFKEEP "1"

/
/ The following constants are used for setting the TCP wait
/ for the select (QZCFWAIT) Environment Variable. The select
/ is used as a timed receive so we do not block forever unless
/ the user wishes to do so.
/
#define ZCFINDEF "-1"
#define ZCFOK " "

/
/ Retail API open flag defines
/
/ The following are defined in FCTNL.H
/
/ O_RDONLY x 1
/ O_WRONLY x 2
/ O_RDWR x 4
/ O_CREAT x 8
/ O_APPEND x 1
/ O_CODEPAGE x 8
/ O_TEXTDATA x 1

/
#define O_FIXED x 1
#define O_MIRRORED x 2
#define O_COMPOUND x 4
#define O_SHARE x 8
#define O_BLOCKL x1
#define O_RECL x2
#define O_KEYED x4
#define O_CCSID x8
#define O_PORT x 1
#define O_ALTLOC x 4
#define O_ALTPORT x 8

#define _NOREPLACE x 1
#define _NOLOCK x 2

/
/
/ START HEADER FILE SPECIFICATIONS
/ File: qzcfpcmn.h
/
/ Description: Retail communications APIs header file
/
/ Product Classification:
/ Retail Application Interface Suite
/ 5X99RAI
/ (C) Copyright IBM Corp. 1999
/ ALL rights reserved.
/ US Government Users Restricted Rights-
/ Use, duplication or disclosure restricted
/ by GSA ADP schedule Contract with IBM Corp.
/
/ Licensed Materials-Property of IBM.
/
/ Function:
/ This module is an AS/4 service program to allow
/ AS/4 applications the ability to open, read, write
/ to the 469 RCO application or the 469 HCP application.
/ The functions include:
/
/ PCMND popen(char ,char ,int, ...);
/ int pcreat(char ,char ,int, ...);
/ int pread(PCMND, void ,int);
/ int pwrite(PCMND, void ,int);
/ long pseek(PCMND, long, int);
/ int pclose(PCMND);
/ int pstat(char ,char ,struct stat ,int, ...);
/ long paccess(char ,char ,int, ...);
/ int punlink(char ,char ,int, ...);
/ int preadk(PCMND, char ,int);
/ int pwritek(PCMND, char ,int, int);
/ int pdeletek(PCMND, char ,int);
/ int pupdatek(PCMND, char ,int, int);
/ int psup(char ,char ,char ,int, ...);
/ long pping(char ,int, ...);
/ short pgetprotocol(char );
/
/
/ Notes: This Module is platform dependent. It is intended
/ for the AS4 .
/
/ Global Return codes:
/
/ The APIs will set the errno variable when an error
/ occurs. These error codes are used to report the API
/ errors which could be system errno values.
/
/ For descriptions of these values see the Retail Application
/ Interface Suite Guide.
/
/ Change Activity: /
/ End Change Activity /

#ifdef _QZCFPCMN_H
#define _QZCFPCMN_H

/
/ includes for common functions
/
#include <errno.h> / C errno definitions
#include <fcntl.h> / open oflag definitions

```

```

/
/
/ Prototyping of functions
/
/

#pragma map(popen,"qzcfopen")
#pragma map(pcreat,"qzcfcreat")
#pragma map(pread,"qzcfread")
#pragma map(pwrite,"qzcfwrite")
#pragma map(pseek,"qzcfseek")
#pragma map(pclose,"qzcfclose")
#pragma map(pstat,"qzcfstat")
#pragma map(paccess,"qzcfaccess")
#pragma map(punlink,"qzcfunlink")
#pragma map(preadk,"qzcfreadk")
#pragma map(pwritek,"qzcfwritek")
#pragma map(pdeletek,"qzcfdeletek")
#pragma map(pupdatek,"qzcfupdatek")
#pragma map(psup,"qzcfsup")
#pragma map(pping,"qzcfping")
#pragma map(pgetprotocol,"qzcfgetprotocol")

QZCF_EXTERN PCMND qzcfopen(char ,char ,int, ...);
QZCF_EXTERN int qzcfcreat(char ,char ,int, ...);
QZCF_EXTERN int qzcfread(PCMND, void ,int);
QZCF_EXTERN int qzcfwrite(PCMND, void ,int);
QZCF_EXTERN long qzcfseek(PCMND, long, int);
QZCF_EXTERN int qzcfclose(PCMND);
QZCF_EXTERN int qzcfstat(char ,char ,struct stat ,int, ...);
QZCF_EXTERN long qzcfaccess(char ,char ,int, ...);
QZCF_EXTERN int qzcfunlink(char ,char ,int, ...);
QZCF_EXTERN int qzcfreadk(PCMND, char ,int);
QZCF_EXTERN int qzcfwritek(PCMND, char ,int, int);
QZCF_EXTERN int qzcfdeletek(PCMND, char ,int);
QZCF_EXTERN int qzcfupdatek(PCMND, char ,int, int);
QZCF_EXTERN int qzcfsup(char ,char ,char ,int, ...);
QZCF_EXTERN long qzcfping(char ,int, ...);

QZCF_EXTERN short qzcfgetprotocol(char );

#endif / _QZCFPCMN_H /

```

pcreat - Create Retail Application File

Syntax

```
#include "QRAIS/h/qzcfpcmn"
int pcreat(char path, char location ,int o_flag, ...);
```

The pcreat() function is exported as qzcfcreate() by the *SRVPGM QZCFPCMN.

The pcreat() function creates a new file on the retail application controller. This can also be used to create a keyed file from a sequential file or create a sequential file from a keyed file on a 4690 controller.

Parameters:

Required Parameters

path (input) A pointer to the null-terminated name of the file to be opened.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

One of the following may be specified:

- Remote 6 character HCP file name,
- Application Logical file name as specified on the retail controller,
- User Logical file name as specified on the retail controller,
- directory/filename.ext.

The directory/filename.ext and user and application logical will be converted to a valid 6 character HCP file name for LU0 if possible. For TCP and LU6.2 protocols, the HCP logical name will be converted to the directory/filename.ext or application/user logical used by the RCO interface.

Note: The format of the path/file should be: *drive:/dir/file.ext* If the drive: is missing C: is the default.

All HCP logical names should be capitalized.

location (input) A pointer to the null-terminated location of the retail application files.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

This can be one of the following:

TCP/IP Host name

Retail(LU0) Remote Location name

Local Integrated File System ('/')

Retail Store Information Store Number

Note: The store number is defined on the Add Retail Store Information screen. This Configuration includes the Primary Location and Secondary Location for a Dual Controller Store. This must be numeric.

o_flag (input) File access open flags.

ex:

open_flag = O_FIXED

open_flag = O_KEYED | O_BLOCKL

o_flag Create Options

O_FIXED **0x0010** : The file is create with fixed length record formats. see the O_RECL parameter. Default is variable length record format.

O_BLOCKL **0x1000** : The file is create with a specified block length. The optional parameter block_length is used, and must be set. The default block length is 0

O_RECL **0x2000** : The file is create with a specified record length. The optional parameter record_length is used, and must be set. The default record_lengths are:

- Fixed Length format: 0
- Variable format: 255
- Keyed Variable format: 253

O_KEYED **0x4000** : The file type to be created is a Keyed file on the 4690. The optional parameters key_length, key_offset, random_divisor and chain_threshold must be specified or unexpected results may occur.

O_MIRRORED **0x0020** : The file type and mode on the HCP create file command is specified as Mirrored.(VOL = 2) Default is Local(VOL = 0). Used for LU0 support only.

O_COMPOUND **0x0040** : The file type and mode on the HCP create file command is specified as Compound.(VOL = 3) Default is Local(VOL = 0). Used for LU0 support only.

O_SHARE **0x0080** : The Share parameter on the HCP create file command is specified as distribute at file close. Default is distribute per update. Used for LU0 support only.

O_PORT **0x010000** : The port of the Retail Connectivity Option TCP server started on the primary retail controller. The optional

parameter `primary_port` is used, and must be set. If this option is not used, the primary port will be defaulted to 20000. Used for TCP/IP support only.

O_ALTLOC

0x040000 : The location of the Retail Connectivity Option server started on the alternate retail controller. The optional parameter `alternate_loc` is used, and must be set. If this option is not used, and the location parameter is not set to a store information store number, no alternate controller support is available.

O_ALTPORT

0x080000 : The port of the Retail Connectivity Option TCP server started on the alternate retail controller. The optional parameter `alternate_port` is used, and must be set. If this option is not used, the alternate port will be defaulted to 20000. Used for TCP/IP support only.

Positional Optional Parameters

- block_length** (int) The length of the file in 256 byte blocks. Must be specified if `O_BLOCKL` is specified. $0 < \text{block_length} < 65535$
- record_length** (int) The specified record length for fixed length records. The Max record length for variable length records. Must be specified if `O_RECL` is specified. For fixed non_keyed files: $\text{record_length} < 256$ For fixed keyed files: $0 < \text{record_length} < 254$ For variable non_keyed files: $0 < \text{record_length} < 255$ For variable keyed files: $0 < \text{record_length} < 253$
- key_length** (int) The length of the Key in a keyed file. Must be specified if `O_KEYED` is specified. For fixed keyed files: $0 < \text{key_length} < 254$ or variable keyed files: $0 < \text{key_length} < 253$
- key_offset** (int) The offset to the Key. Must be $<$ `Record_length`. Must be specified if `O_KEYED` is specified.
- random_divisor** (int) The randomizing divisor. Should be an odd number less than the data set length. Must be specified if `O_KEYED` is specified. $0 < \text{random_divisor} < 65535$
- chain_threshold** (int) The chaining threshold. Should be $>$ 0 Suggested value = 4 Must be specified if `O_KEYED` is specified. $1 < \text{chain_threshold} < 9$
- primary_port** (int) The port for the primary location. Should be between 0 and $<$ 65535 Must be specified if `O_PORT` is specified.
- alternate_loc** (char *) A pointer to the null-terminated name of the alternate location to use. This parameter is assumed to be repre-

sented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

This value should be set to the alternate controller if using dual controller support. Must be specified if `O_ALTLOC` is specified.

alternate_port

(int) The port for the alternate location Should be between 0 and $<$ 65535 Must be specified if `O_ALTPORT` is specified.

Return Value:

- 0** pcreat() was successful. The file was created.
- 1** pcreat() was not successful. The `errno` global variable is set to indicate the error.

ERRNO Values:

- ENODEV** **3007** : The location specified was not correct, or a route to the location does not exist.
- EBADNAME** **3014** : The name or path specified is not correct.
- EBADMODE** **3015** : The mode specified was not valid.
- EINVAL** **3021** : A specified parameter was not valid. Check that the positional parameters are specified in the correct order and that the corresponding Flags are specified.
- ENOENT** **3025** : The source 4690 file does not exist. For HCP, it is possible that the name specified could not be converted to an HCP logical name.
- EBUSY** **3029** : The 4690 resource is busy. Some other application is using the file.
- EBADKEYLN** **3044** : The key length or record length specified is not correct.
- EACCES** **3401** : The 4690 server can not access the file.
- ECONNREFUSED** **3425** : A connection to the remote location was refused by the remote location. Check the status of the server on the remote system.
- EHOSTUNREACH** **3429** : A route to the remote host is not available. For LU0(*RTL) make sure the

ETIMEDOUT	configuration objects are in the proper state. 3447 : The remote location did not respond within the timeout period. Check the device status. Check the remote host server.
EEXIST	3457 : The 4690 destination file already exists.
ENOMEM	3460 : No memory is available on the 4690 or no memory available on the AS/400.
ELOCKED	3506 : The 4690 server can not access the file.

Usage Notes:

See the "popen - Open Retail Application File" on page 3-12 for information regarding resource authorization and locking.

The optional positional parameters must be included in the specified order on the function call when the

corresponding o_flag is set. Do not include optional positional parameters if the corresponding o_flag is not set. In both cases, unexpected results may occur.

The ADCS command sent to the HCP server(for location type ZCFSNA) is the CREATE FILE ('06') command. For keyed files, this only creates the necessary space on the controller. The Keyed File Utility must be run on the controller to convert the file to a keyed file.

The Retail Connectivity Option servers do not make use the O_FIXED, O_MIRRORED, O_COMPOUND and O_SHARE functions on a create. The 4690 File Utilities should be used for these options.

For HCP Logical path conversion for RCO servers, the QZCFAPPL is converted to the 037 CCSID and imbedded into the RCO path. This path is converted to the CCSID specified on the retail_ccsid for use by RCO.

For HCP Logical path conversion for the HCP server the path is converted using the Default Job CCSID. The HCP server must be configured correctly to recognise the HCP logical. The path must be convertible to an HCP logical file name.

popen - Open Retail Application File

Syntax

```
#include "QRAIS/h/qzcfpcmn"  
PCMND popen(char path, char location ,int o_flag, ...);
```

The popen() function is exported as qzcfopen() by the *SRVPGM QZCFPCMN.

The popen() function opens a file on a 4690 controller and returns a number called a file descriptor. You can use this file descriptor to refer to the file in subsequent I/O operations such as pread() or pwrite(). In these subsequent operations, the file descriptor is commonly identified by the argument c. Each file opened by a job gets a new file descriptor. popen() positions the file offset (an indicator showing where the next read or write will take place) at the beginning of the file. However, there are options that can change the position (see pseek).

Parameters:

Required Parameters

path (input) A pointer to the null-terminated name of the file to be opened.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

One of the following may be specified:

- Remote 6 character HCP file name,
- Application Logical file name as specified on the retail controller,
- User Logical file name as specified on the retail controller,
- directory/filename.ext.

The directory/filename.ext and user and application logical will be converted to a valid 6 character HCP file name for LU0 if possible. For TCP and LU6.2 protocols, the HCP logical name will be converted to the directory/filename.ext or application/user logical used by the RCO interface.

Note: The format of the path/file should be: *drive:/dir/file.ext* If the drive is missing C: is the default.

All HCP logical names should be capitalized.

location (input) A pointer to the null-terminated location of the retail application files.

This parameter is assumed to be represented in the CCSID (coded character set

identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

This can be one of the following:

TCP/IP Host name

Retail(LU0) Remote Location name

Local Integrated File System ('/')

Retail Store Information Store Number

Note: The store number is defined on the Add Retail Store Information screen. This Configuration includes the Primary Location and Secondary Location for a Dual Controller Store. This must be numeric.

o_flag (input) File access open flags.

ex:

open_flag = O_RDONLY

open_flag = O_WRONLY | O_PORT

o_flag File Access Options

O_KEYED **0x4000** : The file type to be opened is a Keyed file on the 4690.

O_RDONLY **0x0001** : The file is opened for read cannot be specified if O_WRONLY or O_RDWR are specified.

O_WRONLY **0x0002** : The file is opened for write cannot be specified if O_RDONLY or O_RDWR are specified.

O_RDWR **0x0004** : The file is opened for write and read cannot be specified if O_WRONLY or O_RDONLY are specified.

O_APPEND **0x0100** : The file is opened for write and data will be appended to the end of the file. If not specified, the file will be overwritten. O_WRONLY or O_RDWR must be specified.

O_CREAT **0x0008** : The file is created if it does not exist. See Create options. This cannot be specified if O_RDWR or O_RDONLY are specified. This also can not be specified for KEYED files. KEYED files must be created using the pcreat() function.

O_TEXTDATA **0x01000000** : The file is opened for write or read and the data is treated as Character data and will be converted between the retail CCSID and the AS/400 Job Default CCSID. If the O_CCSID option is used, this value will be used as the retail CCSID.

O_CCSID **0x8000** : This is the CCSID of all the retail application data that is accessed through this API. This includes file data

and file names. The optional parameter `retail_ccsid` is used, and must be set.

- O_PORT** **0x010000** : The port of the Retail Connectivity Option TCP server started on the primary retail controller. The optional parameter `primary_port` is used, and must be set. If this option is not used, the primary port will be defaulted to 20000. Used for TCP/IP support only.
- O_ALTLOC** **0x040000** : The location of the Retail Connectivity Option server started on the alternate retail controller. The optional parameter `alternate_loc` is used, and must be set. If this option is not used, and the location parameter is not set to a store information store number, no alternate controller support is available.
- O_ALTPORT** **0x080000** : The port of the Retail Connectivity Option TCP server started on the alternate retail controller. The optional parameter `alternate_port` is used, and must be set. If this option is not used, the alternate port will be defaulted to 20000. Used for TCP/IP support only.

o_flag Create Options

- O_FIXED** **0x0010** : The file is create with fixed length record formats. see the `O_RECL` parameter. Default is variable length record format.
- O_BLOCKL** **0x1000** : The file is create with a specified block length. The optional parameter `block_length` is used, and must be set. The default block length is 0
- O_RECL** **0x2000** : The file is create with a specified record length. The optional parameter `record_length` is used, and must be set. The default record_lengths are:
- Fixed Length format: 0
 - Variable format: 255
 - Keyed Variable format: 253
- O_MIRRORED** **0x0020** : The file type and mode on the HCP create file command is specified as Mirrored.(VOL = 2) Default is Local(VOL = 0). Used for LU0 support only.
- O_COMPOUND** **0x0040** : The file type and mode on the HCP create file command is specified as Compound.(VOL = 3) Default is Local(VOL = 0). Used for LU0 support only.
- O_SHARE** **0x0080** : The Share parameter on the HCP create file command is specified as distribute at file close. Default is distribute per update. Used for LU0 support only.

Positional Optional Parameters

- block_length** (*int*) The length of the file in 256 byte blocks. Must be specified if `O_BLOCKL` is specified. $0 < \text{block_length} < 65535$
- record_length** (*int*) The specified record length for fixed length records. The Max record length for variable length records. Must be specified if `O_RECL` is specified. For fixed non_keyed files: $\text{record_length} < 256$ For fixed keyed files: $0 < \text{record_length} < 254$ For variable non_keyed files: $0 < \text{record_length} < 255$ For variable keyed files: $0 < \text{record_length} < 253$
- retail_ccsid** (*int*) The ccsid number of the retail data to be converted. Should be between 0 and < 65535 If 0 or a number > 65000 is specified, the default 850 CCSID will be used in conversion. Must be specified if `O_CCSD` is specified.
- primary_port** (*int*) The port for the primary location Should be between 0 and < 65535 Must be specified if `O_PORT` is specified.
- alternate_loc** (*char **) A pointer to the null-terminated name of the alternate location to use.
- This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.
- This value should be set to the alternate controller if using dual controller support. Must be specified if `O_ALTLOC` is specified.
- alternate_port** (*int*) The port for the alternate location Should be between 0 and < 65535 Must be specified if `O_ALTPORT` is specified.

Return Value:

- PCMND value** `popen()` was successful. A valid retail application filed descriptor pointer is returned.
- NULL** `popen()` was not successful. The `errno` global variable is set to indicate the error.

ERRNO Values:

- ENODEV** **3007** : The location specified was not correct, or a route to the location does not exist.
- EBADNAME** **3014** : The name or path specified is not correct.
- EBADMODE** **3015** : The mode specified was not valid.
- EINVAL** **3021** : A specified parameter was not valid. Check that the positional parameters are

specified in the correct order and that the corresponding Flags are specified.

ENOENT **3025** : The source 4690 file does not exist. For HCP, it is possible that the name specified could not be converted to an HCP logical name.

EBUSY **3029** : The 4690 resource is busy. Some other application is using the file.

EBADKEYLN **3044** : The key length or record length specified is not correct.

EACCES **3401** : The 4690 server can not access the file.

ECONNREFUSED **3425** : A connection to the remote location was refused by the remote location. Check the status of the server on the remote system.

EHOSTUNREACH **3429** : A route to the remote host is not available. For LU0(*RTL) make sure the configuration objects are in the proper state.

ERCVDERR **3444** : The remote application sent an application error notification. Check the 4690 OS messages to see why the remote application could not access files on its local area network. This error occurs when there is no file server running in a Multi-Controller 4690 system.

ETIMEDOUT **3447** : The remote location did not respond within the timeout period. Check the device status. Check the remote host server.

ENOMEM **3460** : No memory is available on the 4690 or no memory available on the AS/400.

ELOCKED **3506** : The 4690 server can not access the file.

EBADFID **3512** : The Keyed file specified is not in the correct state for processing.

Usage Notes:

The positional optional parameters must be included in the specified order on the function call when the corresponding o_flag is set. Do not include optional posi-

tional parameters if the corresponding o_flag is not set. In both cases, unexpected results may occur.

For HCP Logical path conversion for RCO servers, the QZCFAPPL is converted to the 037 CCSID and imbedded into the RCO path. This path is converted to the CCSID specified on the retail_ccsid for use by RCO.

For HCP Logical path conversion for the HCP server the path is converted using the Default Job CCSID. The HCP server must be configured correctly to recognize the HCP logical. The path must be convertible to an HCP logical file name.

Authorization: File and directory access is handled by each of the Retail Application servers. EACCESS will be returned by the API when the Servers can not access the requested resource.

Location Type ZCFSNA

The Host Command Processor will signal the following negative responses to the API.

00003FC5

00003F9D

Note: HCP does not allow write access to the ADX_SPGM and ADX_IPGM directories. See the *IBM 4690 Store System Communications Programming Reference* for additional HCP information. See the *IBM 4690 Store System Messages Guide* for a description of the HCP sense codes.

Location Type ZCFIP

The Retail Connectivity Option(RCO) servers will return the following error codes:

80004001

80F34001

8000400C

80F3400C

8020400C

80F306CF

80F306F0

80204306

Note: The RCO errors returned are the 4690 Return Codes. See the *Retail Connectivity Option Product Reference* for additional RCO authorization information. Also see the *IBM 4690 Store System Messages Guide* for a description of the 4690 Return Codes.

Location Type ZCFLOCAL

See the open() function documented in the *System API Reference OS/400 UNIX-Type APIs* for information regarding authorizations to different file systems.

Locking

For the ZCFSNA support, the location can be locked by another AS/400 process. In this case EBUSY will be returned. To allow for multiple processes to share One

*RTL location, set the QZCFCNV Environment variable to ZCFDROP.

All other locking on the files is handled by the Retail Application servers.

pseek - Set Retail Application File Read/Write Offset

Syntax

```
#include "QRAIS/h/qzcfpcmn"  
long pseek(PCMND fildes, long offset, int whence);
```

The pseek() function is exported as qzcfseek() by the *SRVPGM QZCFPCMN.

The pseek() function changes the current file offset to a new position in the file. The new position is the given byte offset from the position specified by offset. After you have used pseek() to seek to a new location, the next I/O operation on the file begins at that location. pseek() lets you specify new file offsets past the current end of the file. A pread() at this point will return 0 bytes read. A pwrite() will return a bad offset error.

Parameters:

Required Parameters

fildes	(input) The file descriptor which was returned on the open.
offset	(input) The amount (positive or negative) the byte offset is to be changed. The sign indicates whether the offset is to be moved forward (positive) or backward(negative).
whence	(input) One of the following symbols, starting point of the seek.

SEEK_SET	0 : The offset is the displacement(+) from the start of the file
SEEK_CUR	1 : The offset is the displacement(+,-) from the current file pointer.
SEEK_END	2 : The offset is the displacement(-) from the end of the file.

Return Value:

value	pseek() was successful. The file offset is now set to the value returned.
-1	pseek() was not successful. The errno global variable is set to indicate the error.

ERRNO Values:

ENOTOPEN	3004 : The file descriptor is referencing a closed file.
EBADSEEK	3013 : The offset is referencing a displacement in the file that is not valid.
ECONNABORTED	3424 : A communication error occurred that caused the connection to drop.
EBADF	3450 : The file descriptor used is not valid.

pread - Read from Retail Application File Descriptor

Syntax

```
#include "QRAIS/h/qzcfpcmn"  
int pread(PCMND fildes, void buf, int nbyte);
```

The pread() function is exported as qzcfread() by the *SRVPGM QZCFPCMN.

From the file indicated by fildes, the pread() function reads nbyte bytes of input into the memory area indicated by buf. If nbyte is zero, pread() returns a value of zero without attempting any other action.

pread() will set the file offset to the end of the data which was read.

If fildes refers to a descriptor obtained using the popen() function with O_TEXTDATA specified, the data is read from the file assuming it is in textual form. The data is converted from the code page of the file(850) to the code page of the application, or job.

If O_TEXTDATA was not specified on the popen(), the data is read from the file without conversion. The application is responsible for handling the data.

Parameters:

Required Parameters

fildes (input) The file descriptor which was returned on the open.

buf (input) Pointer to the buffer in which the read function will place the data read.

nbyte (input) The buffer size.

Return Value:

value = nbyte pread() was successful. The value returned is the number of bytes that were read.

value < nbyte, errno = 0 pread() was successful.

Note: EOF is determined if the bytes_read < nbyte and errno is 0.

value < nbyte, errno > 0 pread() was not successful. The errno global variable is set to indicate the error.

ERRNO Values:

ENOTOPEN

3004 : The file descriptor is referencing a closed file.

ENOTREAD

3005 : The file descriptor is referencing a file that has not been opened for read.

EIO

3006 : A communications IO error occurred. Generally more data was returned than was expected, or the link went down.

ECONNABORTED

3424 : A communication error occurred that caused the connection to drop.

ETIMEDOUT

3447 : The remote location did not respond within the timeout period. This may not be an error condition. You may issue another read operation, or issue pclose() to close the file descriptor.

EBADF

3450 : The file descriptor used is not valid.

ETERM

3464 : A Communications return code was returned that was not expected, so the connection was dropped.

pwrite - Write to Retail Application File Descriptor

Syntax

```
#include "QRAIS/h/qzcfpcmn"  
int pwrite(PCMND fildes, void buf, int nbyte);
```

The pwrite() function is exported as qzcfwrite() by the *SRVPGM QZCFPCMN.

The pwrite() function writes nbyte bytes from buf to the retail file associated with fildes. nbyte should not be greater than INT_MAX (defined in the <limits.h> header file). If nbyte is zero, pwrite() simply returns a value of zero without attempting any other action.

pwrite() will set the file offset to the end of the data which was written.

If fildes refers to a descriptor obtained using the popen() function with O_TEXTDATA specified, the data is written to the file assuming it is in textual form. The data is converted to the code page of the file(850) from the code page of the application, or job.

If O_TEXTDATA was not specified on the popen(), the data is written from the file without conversion. The application is responsible for handling the data.

Parameters:

Required Parameters

fildes	(input) The file descriptor which was returned on the open.
buf	(input) Pointer to the buffer in which the application places the data to be written for the write function.
nbyte	(input) The data size.

Return Value:

value pwrite() was successful. The value returned is the number of bytes that were written.

-1 pwrite() was not successful. The errno global variable is set to indicate the error.

ERRNO Values:

ENOTOPEN

3004 : The file descriptor is referencing a closed file.

EIO

3006 : A communications IO error occurred. Generally more data was returned than was expected, or the link went down.

ENOTWRITE

3009 : The file descriptor is referencing a file that has not been opened for write.

ECONNABORTED

3424 : A communication error occurred that caused the connection to drop.

ETIMEDOUT

3447 : The remote location did not respond within the timeout period. This may not be an error condition. You may issue another write operation, or issue pclose() to close the file descriptor.

EBADF

3450 : The file descriptor used is not valid.

ETERM

3464 : A Communications return code was returned that was not expected, so the connection was dropped.

preadk - Read from Keyed Retail Application File Descriptor

Syntax

```
#include "QRAIS/h/qzcfpcmn"
int preadk(PCMND fildes, void buf, int nbyte);
```

The preadk() function is exported as qzcfreadk() by the *SRVPGM QZCFPCMN.

From the file indicated by fildes, the preadk() function reads nbyte bytes of input into the memory area indicated by buf. buf is also used as an input parameter which holds the record key to be found.

Prototype:: int preadk(PCMND, void *,int);

Invocation:: bytes_read preadk(fildes,buf,nbyte);

Parameters:

Required Parameters

fildes	(input) The file descriptor which was returned on the open.
buf	(input/output) Pointer to the buffer in which the key of the record is placed. This is also the buffer which the complete record is read into.
nbyte	(input) The buffer size of the buf parameter. If this buffer size is less than the Key length of the retail system file an EBADKEYLN errno is returned.

Return Value:

value = nbyte preadk() was successful. The value returned is the number of bytes that were read.

value < nbyte, errno = 0 preadk() was successful.

Note: EOF is determined if the bytes_read < nbyte and errno is 0.

value < nbyte, errno > 0 preadk() was not successful. The errno global variable is set to indicate the error.

ERRNO Values:

ENOTOPEN

3004 : The file descriptor is referencing a closed file.

ENOTREAD

3005 : The file descriptor is referencing a file that has not been opened for read.

EIO

3006 : A communications IO error occurred. Generally more data was returned than was expected, or the link went down.

ENOREC

3026 : The specified record key was not found in the retail application file.

EBADKEYLN

3044 : The record length specified is not correct.

ECONNABORTED

3424 : A communication error occurred that caused the connection to drop.

ETIMEDOUT

3447 : The remote location did not respond within the timeout period. This may not be an error condition. You may issue another read operation, or issue pclose() to close the file descriptor.

EBADF

3450 : The file descriptor used is not valid.

ETERM

3464 : A Communications return code was returned that was not expected, so the connection was dropped.

Usage Notes:

The key length and record length of the Retail files on 4690 Retail Controllers can be determined by using the Keyed File Utilities. See the *IBM 4690 Store System: Programmers Guide* for more information on using the Keyed File Utility.

pwritek - Write to Keyed Retail Application File Descriptor

Syntax

```
#include "QRAIS/h/qzcfpcmn"
int pwritek(PCMND fildes, void buf, int nbyte, int options);
```

The pwritek() function is exported as qzcfwritek() by the *SRVPGM QZCFPCMN.

The pwritek() function writes nbyte bytes from buf to the retail file associated with fildes. The buf parameter should hold the record to be written to the file.

Parameters:

Required Parameters

fildes	(input) The file descriptor which was returned on the open.
buf	(input) Pointer to the buffer in which the application places the data to be written for the write function.
nbyte	(input) The size of the buf data. If this size is less than the retail system file record length an EBADKEYLN errno is returned. Note: THIS IS NOT THE KEY LENGTH.
options	(input) The options on the write keyed record. The only valid option is <code>_NOREPLACE (0x0001)</code> : the record should not be replaced if it exists. If you wish to replace, use the pupdatek() function.

Return Value:

value	pwritek() was successful. The value returned is the number of bytes that were written.
--------------	--

-1

pwritek() was not successful. The errno global variable is set to indicate the error.

ERRNO Values:

ENOTOPEN

3004 : The file descriptor is referencing a closed file.

EIO

3006 : A communications IO error occurred. Generally more data was returned than was expected, or the link went down.

ENOTWRITE

3009 : The file descriptor is referencing a file that has not been opened for write.

EBADKEYLN

3044 : The record length specified is not correct.

ECONNABORTED

3424 : A communication error occurred that caused the connection to drop.

ETIMEDOUT

3447 : The remote location did not respond within the timeout period. This may not be an error condition. You may issue another write operation, or issue pclose() to close the file descriptor.

EBADF

3450 : The file descriptor used is not valid.

ETERM

3464 : A Communications return code was returned that was not expected, so the connection was dropped.

Usage Notes:

The key length and record length of the Retail files on 4690 Retail Controllers can be determined by using the Keyed File Utilities. See the *IBM 4690 Store System: Programmers Guide* for more information on using the Keyed File Utility.

pupdatek - Update to Keyed Retail Application File Descriptor

Syntax

```
#include "QRAIS/h/qzcfpcmn"
int pupdatek(PCMND fildes, void buf, int nbyte, int options);
```

The pupdatek() function is exported as qzcfupdatek() by the *SRVPGM QZCFPCMN.

The pupdatek() function writes nbyte bytes from buf to the retail file associated with fildes. The buf parameter should hold the record to be updated in the file.

Required Parameters

fildes	(input) The file descriptor which was returned on the open.
buf	(input) Pointer to the buffer in which the application places the data to be written for the write function.
nbyte	(input) The size of the buf data. If this size is less than the retail system file record length an EBADKEYLN errno is returned. Note: THIS IS NOT THE KEY LENGTH.
options	(input) The options on the update keyed record. The only valid option is _NOLOCK (0x0001) : the record should be locked on update if it exists.

Return Value:

value	pupdatek() was successful. The value returned is the number of bytes that were written.
-1	pupdatek() was not successful. The errno global variable is set to indicate the error.

ERRNO Values:

ENOTOPEN

EIO

ENOTWRITE

ENOREC

EBADKEYLN

ECONNABORTED

ETIMEDOUT

EBADF

ETERM

Usage Notes:

The key length and record length of the Retail files on 4690 Retail Controllers can be determined by using the Keyed File Utilities. See the *IBM 4690 Store System: Programmers Guide* for more information on using the Keyed File Utility.

3004: The file descriptor is referencing a closed file.

3006: A communications IO error occurred. Generally more data was returned than was expected, or the link went down.

3009: The file descriptor is referencing a file that has not been opened for write.

3026: The specified record key was not found in the retail application file.

3044: The record length specified is not correct.

3424: A communication error occurred that caused the connection to drop.

3447: The remote location did not respond within the timeout period. This may not be an error condition. You may issue another update operation, or issue pclose() to close the file descriptor.

3450: The file descriptor used is not valid.

3464: A Communications return code was returned that was not expected, so the connection was dropped.

pdeletek - Delete to Keyed Retail Application File Descriptor

Syntax

```
#include "QRAIS/h/qzcfpcmn"  
int pdeletek(PCMND fildes, void buf, int nbyte);
```

The pdeletek() function is exported as qzcfdeletek() by the *SRVPGM QZCFPCMN.

The pdeletek() function will delete the record specified in buf from the retail file associated with fildes.

Parameters:

Required Parameters

fildes (input) The file descriptor which was returned on the open.

buf (input) Pointer to the buffer in which the application places the key data to be used in the delete function.

nbyte (input) The length of the keyed data. If this data length is less than the Key length of the retail system file an EBADKEYLN errno is returned.

Return Value:

0 pdeletek() was successful. The record was deleted.

-1 pdeletek() was not successful. The errno global variable is set to indicate the error.

ERRNO Values:

ENOTOPEN **3004** : The file descriptor is referencing a closed file.

EIO

3006 : A communications IO error occurred. Generally more data was returned than was expected, or the link went down.

ENOTWRITE

3009 : The file descriptor is referencing a file that has not been opened for write.

ENOREC

3026 : The specified record key was not found in the retail application file.

EBADKEYLN

3044 : The record length specified is not correct.

ECONNABORTED

3424 : A communication error occurred that caused the connection to drop.

ETIMEDOUT

3447 : The remote location did not respond within the timeout period. This may not be an error condition. You may issue another delete operation, or issue pclose() to close the file descriptor.

EBADF

3450 : The file descriptor used is not valid.

ETERM

3464 : A Communications return code was returned that was not expected, so the connection was dropped.

Usage Notes:

The key length and record length of the Retail files on 4690 Retail Controllers can be determined by using the Keyed File Utilities. See the *IBM 4690 Store System: Programmers Guide* for more information on using the Keyed File Utility.

pclose - Close Retail Application File Descriptor

Syntax

```
#include "QRAIS/h/qzcfpcmn"  
int pclose(PCMND fildes);
```

The pclose() function is exported as qzcfclose() by the *SRVPGM QZCFPCMN.

The pclose() function closes a descriptor, fildes. This frees the descriptor and associated memory. pclose() may also close the remote 4690 file and the Communications conversation.

Parameters:

Required Parameters

fildes (input) The file descriptor which was returned on the open.

Return Value:

0 pclose() was successful.

-1 pclose() was not successful. The errno global variable is set to indicate the error.

ERRNO Values:

ENOTOPEN

3004 : The file descriptor is referencing a closed file.

EIO

3006 : A communications IO error occurred. Generally more data was returned than was expected, or the link went down.

ECONNABORTED

3424 : A communication error occurred that caused the connection to drop.

EBADF

3450 : The file descriptor used is not valid.

ETERM

3464 : A Communications return code was returned that was not expected, so the connection was dropped.

pstat - Get Retail Application File Information

Syntax

```
#include "QRAIS/h/qzcfpcmn"
int pstat(char path, char location ,struct stat buf,
int flag, ...);
```

The pstat() function is exported as qzcfstat() by the *SRVPGM QZCFPCMN.

The pstat() function gets status information about a specified Retail Application file and places it in the area of memory pointed to by buf.

Parameters:

Required Parameters

path (input) A pointer to the null-terminated name of the file to be opened.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

One of the following may be specified:

- Remote 6 character HCP file name,
- Application Logical file name as specified on the retail controller,
- User Logical file name as specified on the retail controller,
- directory/filename.ext.

The directory/filename.ext and user and application logical will be converted to a valid 6 character HCP file name for LU0 if possible. For TCP and LU6.2 protocols, the HCP logical name will be converted to the directory/filename.ext or application/user logical used by the RCO interface.

Note: The format of the path/file should be: *drive:/dir/file.ext* If the drive: is missing C: is the default.

Note: For HCP locations, application and user logical names should not be used. The only directory searched will be the root('\') directory.

All HCP logical names should be capitalized.

location (input) A pointer to the null-terminated location of the retail application files.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this param-

eter is assumed to be represented in the default CCSID of the job.

This can be one of the following:

- TCP/IP Host name
- Retail(LU0) Remote Location name
- Local Integrated File System ('/')
- Retail Store Information Store Number

Note: The store number is defined on the Add Retail Store Information screen. This Configuration includes the Primary Location and Secondary Location for a Dual Controller Store. This must be numeric.

buf (output) The stat structure which holds the file attribute information. HCP must read all the files in the sub-directory to get this information.

flag (input) optional parameter flags.

ex:

```
stat_flag = O_PORT
stat_flag = O_ALTLOC | O_ALTPORT
```

O_PORT The port of the Retail Connectivity Option TCP server started on the primary retail controller. The optional parameter primary_port is used, and must be set. If this option is not used, the primary port will be defaulted to 20000. Used for TCP/IP support only.

O_ALTLOC The location of the Retail Connectivity Option server started on the alternate retail controller. The optional parameter alternate_loc is used, and must be set. If this option is not used, and the location parameter is not set to a store information store number, no alternate controller support is available.

O_ALTPORT The port of the Retail Connectivity Option TCP server started on the alternate retail controller. The optional parameter alternate_port is used,

and must be set. If this option is not used, the alternate port will be defaulted to 20000. Used for TCP/IP support only.

Positional Optional Parameters

- primary_port** (int) The port for the primary location. Should be between 0 and < 65535 Must be specified if O_PORT is specified.
- alternate_loc** (char *) A pointer to the null-terminated name of the alternate location to use.
 This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.
 This value should be set to the alternate controller if using dual controller support. Must be specified if O_ALTLOC is specified.
- alternate_port** (int) The port for the alternate location. Should be between 0 and < 65535 Must be specified if O_ALTPORT is specified.

Return Value:

- 0** pstat() was successful. The record was deleted.
- 1** pstat() was not successful. The errno global variable is set to indicate the error.

ERRNO Values:

- ENODEV** **3007** : The location specified was not correct, or a route to the location does not exist.
- EBADNAME** **3014** : The name or path specified is not correct.
- ENOENT** **3025** : The source 4690 file does not exist. For HCP, it is possible that the name specified could not be converted to an HCP logical name.

- EBUSY** **3029** : The 4690 resource is busy. Some other application is using the file.
- EACCES** **3401** : The 4690 server can not access the file.
- ECONNREFUSED** **3425** : A connection to the remote location was refused by the remote location. Check the status of the server on the remote system.
- EHOSTUNREACH** **3429** : A route to the remote host is not available. For LU0(*RTL) make sure the configuration objects are in the proper state.
- ETIMEDOUT** **3447** : The remote location did not respond within the timeout period. Check the device status. Check the remote host server.
- ENOMEM** **3460** : No memory is available on the 4690 or no memory available on the AS/400.
- ELOCKED** **3506** : The 4690 server can not access the file.

Usage Notes:

See the "popen - Open Retail Application File" on page 3-12 for information regarding resource authorization and locking.

For HCP Logical path conversion for RCO servers, the QZCFAPPL is converted to the 037 CCSID and imbedded into the RCO path. This path is converted to the CCSID specified on the retail_ccsid for use by RCO.

For HCP Logical path conversion for the HCP server the path is converted using the Default Job CCSID. The HCP server must be configured correctly to recognise the HCP logical. The path must be convertible to an HCP logical file name.

This function will not work with User Defined Logical names when connected with the HCP server. The pstat() function for *RTL location types uses the EXTENDED DUMP(x'0303)' command to retrieve the directory the file exists in, then searches the directory dump for the specified file. For User Defined Logical names, the directory is unknown to the API.

passcess - Check Retail Application File Access

Syntax

```
#include "QRAIS/h/qzcfpcmn"  
int passcess(char path, char location, int flag, ...);
```

The passcess() function is exported as qzcfaccess() by the *SRVPGM QZCFPCMN.

The passcess() function determines whether a Retail application file exists. If it does passcess() will return the file size back to the program.

Parameters:

Required Parameters

path (input) A pointer to the null-terminated name of the file to be opened.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

One of the following may be specified:

- Remote 6 character HCP file name,
- Application Logical file name as specified on the retail controller,
- User Logical file name as specified on the retail controller,
- directory/filename.ext.

The directory/filename.ext and user and application logical will be converted to a valid 6 character HCP file name for LU0 if possible. For TCP and LU6.2 protocols, the HCP logical name will be converted to the directory/filename.ext or application/user logical used by the RCO interface.

Note: The format of the path/file should be: *drive:/dir/file.ext* If the drive: is missing C: is the default.

Note: For HCP locations, application and user logical names should not be used. The only directory searched will be the root("\") directory.

All HCP logical names should be capitalized.

location (input) A pointer to the null-terminated location of the retail application files.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this param-

eter is assumed to be represented in the default CCSID of the job.

This can be one of the following:

- TCP/IP Host name
- Retail(LU0) Remote Location name
- Local Integrated File System ('/')
- Retail Store Information Store Number

Note: The store number is defined on the Add Retail Store Information screen. This Configuration includes the Primary Location and Secondary Location for a Dual Controller Store. This must be numeric.

(input) optional parameter flags.

ex:

```
acc_flag = O_PORT  
acc_flag = O_ALTLOC | O_ALTPORT
```

O_PORT The port of the Retail Connectivity Option TCP server started on the primary retail controller. The optional parameter primary_port is used, and must be set. If this option is not used, the primary port will be defaulted to 20000. Used for TCP/IP support only.

O_ALTLOC The location of the Retail Connectivity Option server started on the alternate retail controller. The optional parameter alternate_loc is used, and must be set. If this option is not used, and the location parameter is not set to a store information store number, no alternate controller support is available.

O_ALTPORT The port of the Retail Connectivity Option TCP server started on the alternate retail controller. The optional parameter alternate_port is used, and must be set. If this option is not used, the alternate port will be defaulted to 20000.

Used for TCP/IP support only.

Positional Optional Parameters

- primary_port** (*int*) The port for the primary location. Should be between 0 and < 65535 Must be specified if O_PORT is specified.
- primary_mode** (*char **) A pointer to the null-terminated name of the mode to use primary location. This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job. Should be between a defined mode on the AS/400 and the Retail Controller. Must be specified if O_MODE is specified.
- alternate_loc** (*char **) A pointer to the null-terminated name of the alternate location to use. This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job. This value should be set to the alternate controller if using dual controller support. Must be specified if O_ALTLOC is specified.
- alternate_port** (*int*) The port for the alternate location. Should be between 0 and < 65535 Must be specified if O_ALTPORT is specified.
- alternate_mode** (*char **) A pointer to the null-terminated name of the mode to use for the alternate location. This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job. Should be between a defined mode on the AS/400 and the Retail Controller. Must be specified if O_ALTMODE is specified.

Return Value:

- value** paccess() was successful. The value returned is the retail file size.
- 1** paccess() was not successful. The errno global variable is set to indicate the error.

ERRNO Values:

- ENODEV** **3007** : The location specified was not correct, or a route to the location does not exist.
- EBADNAME** **3014** : The name or path specified is not correct.
- ENOENT** **3025** : The source 4690 file does not exist. For HCP, it is possible that the name specified could not be converted to an HCP logical name.
- EBUSY** **3029** : The 4690 resource is busy. Some other application is using the file.
- EACCES** **3401** : The 4690 server can not access the file.
- ECONNREFUSED** **3425** : A connection to the remote location was refused by the remote location. Check the status of the server on the remote system.
- EHOSTUNREACH** **3429** : A route to the remote host is not available. For LU0(*RTL) make sure the configuration objects are in the proper state.
- ETIMEDOUT** **3447** : The remote location did not respond within the timeout period. Check the device status. Check the remote host server.
- ENOMEM** **3460** : No memory is available on the 4690 or no memory available on the AS/400.
- ELOCKED** **3506** : The 4690 server can not access the file.

Usage Notes:

See the "popen - Open Retail Application File" on page 3-12 for information regarding resource authorization and locking.

For HCP Logical path conversion for RCO servers, the QZCFAPPL is converted to the 037 CCSID and imbedded into the RCO path. This path is converted to the CCSID specified on the retail_ccsid for use by RCO.

For HCP Logical path conversion for the HCP server the path is converted using the Default Job CCSID. The HCP server must be configured correctly to recognize the HCP logical. The path must be convertible to an HCP logical file name.

This function will not work with User Defined Logical names when connected with the HCP server. The paccess() function for *RTL location types uses the EXTENDED DUMP(x'0303)' command to retrieve the directory the file exists in, then searches the directory

dump for the specified file. For User Defined Logical names, the directory is unknown to the API.

punlink - Delete Retail Application File

Syntax

```
#include "QRAIS/h/qzcfpcmn"
int punlink(char path, char location ,int flag, ...);
```

The punlink() function is exported as qzcfunlink() by the *SRVPGM QZCFPCMN.

The punlink() function removes the specified Retail application file.

Parameters:

Required Parameters

path (input) A pointer to the null-terminated name of the file to be opened.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

One of the following may be specified:

- Remote 6 character HCP file name,
- Application Logical file name as specified on the retail controller,
- User Logical file name as specified on the retail controller,
- directory/filename.ext.

The directory/filename.ext and user and application logical will be converted to a valid 6 character HCP file name for LU0 if possible. For TCP and LU6.2 protocols, the HCP logical name will be converted to the directory/filename.ext or application/user logical used by the RCO interface.

Note: The format of the path/file should be: *drive:/dir/file.ext* If the drive is missing C: is the default.

All HCP logical names should be capitalized.

location (input) A pointer to the null-terminated location of the retail application files.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

This can be one of the following:

- TCP/IP Host name
- Retail(LU0) Remote Location name

Local Integrated File System ('/')

Retail Store Information Store Number

Note: The store number is defined on the Add Retail Store Information screen. This Configuration includes the Primary Location and Secondary Location for a Dual Controller Store. This must be numeric.

flag

(input) optional parameter flags.

ex:

```
acc_flag = O_PORT
```

```
acc_flag = O_ALTLOC | O_ALTPORT
```

O_PORT

The port of the Retail Connectivity Option TCP server started on the primary retail controller. The optional parameter primary_port is used, and must be set. If this option is not used, the primary port will be defaulted to 20000. Used for TCP/IP support only.

O_ALTLOC

The location of the Retail Connectivity Option server started on the alternate retail controller. The optional parameter alternate_loc is used, and must be set. If this option is not used, and the location parameter is not set to a store information store number, no alternate controller support is available.

O_ALTPORT

The port of the Retail Connectivity Option TCP server started on the alternate retail controller. The optional parameter alternate_port is used, and must be set. If this option is not used, the alternate port will be defaulted to 20000. Used for TCP/IP support only.

Positional Optional Parameters

primary_port (*int*) The port for the primary location. Should be between 0 and < 65535 Must be specified if O_PORT is specified.

alternate_loc (*char **) A pointer to the null-terminated name of the alternate location to use.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

This value should be set to the alternate controller if using dual controller support. Must be specified if O_ALTLOC is specified.

alternate_port (*int*) The port for the alternate location. Should be between 0 and < 65535. Must be specified if O_ALTPORT is specified.

Return Value:

- 0** punlink() was successful. The file was deleted.
- 1** punlink() was not successful. The errno global variable is set to indicate the error.

ERRNO Values:

- ENODEV** **3007** : The location specified was not correct, or a route to the location does not exist.
- EBADNAME** **3014** : The name or path specified is not correct.
- ENOENT** **3025** : The source 4690 file does not exist. For HCP, it is possible that the name specified could not be converted to an HCP logical name.
- EBUSY** **3029** : The 4690 resource is busy. Some other application is using the file.

EACCES

ECONNREFUSED

EHOSTUNREACH

ETIMEDOUT

ENOMEM

ELOCKED

3401 : The 4690 server can not access the file.

3425 : A connection to the remote location was refused by the remote location. Check the status of the server on the remote system.

3429 : A route to the remote host is not available. For LU0(*RTL) make sure the configuration objects are in the proper state.

3447 : The remote location did not respond within the timeout period. Check the device status. Check the remote host server.

3460 : No memory is available on the 4690 or no memory available on the AS/400.

3506 : The 4690 server can not access the file.

Usage Notes:

See the "popen - Open Retail Application File" on page 3-12 for information regarding resource authorization and locking.

For HCP Logical path conversion for RCO servers, the QZCFAPPL is converted to the 037 CCSID and imbedded into the RCO path. This path is converted to the CCSID specified on the retail_ccsid for use by RCO.

For HCP Logical path conversion for the HCP server the path is converted using the Default Job CCSID. The HCP server must be configured correctly to recognize the HCP logical. The path must be convertible to an HCP logical file name.

psup - Start Retail Application Program

Syntax

```
#include "QRAIS/h/qzcfpcmn"
int psup(char path, char location, char parms, int flag, ...);
```

The psup() function is exported as qzcfpsup() by the *SRVPGM QZCFPCMN.

The psup() function starts a Retail Application program specified in the path.

Parameters:

Required Parameters

path (input) A pointer to the null-terminated name of the program to be started.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

This can be: Remote 6 character HCP program name or directory/program.ext. The directory/program.ext will be converted to a valid 6 character HCP file name for LU0. For TCP and LU6.2 protocols, the HCP logical name will be converted to the directory/program.ext used by the RCO interface.

Note: The format of the path/file should be: *drive:/dir/program.ext* If the drive: is missing C: is the default.

All HCP logical names should be capitalized.

location (input) A pointer to the null-terminated location of the retail application files.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

This can be one of the following:

- TCP/IP Host name
- Retail(LU0) Remote Location name
- Local Integrated File System ('/')
- Retail Store Information Store Number

Note: The store number is defined on the Add Retail Store Information screen. This Configuration includes the Primary Location and

parms

flag

Secondary Location for a Dual Controller Store. This must be numeric.

Note: For LU0 support the start user program will only be started over the SUP defined Retail Device.

(input) The program parameters.

(input) optional parameter flags.

ex:

```
acc_flag = O_PORT
```

```
acc_flag = O_ALTLOC | O_ALTPORT
```

O_PORT

The port of the Retail Connectivity Option TCP server started on the primary retail controller. The optional parameter primary_port is used, and must be set. If this option is not used, the primary port will be defaulted to 20000. Used for TCP/IP support only.

O_ALTLOC

The location of the Retail Connectivity Option server started on the alternate retail controller. The optional parameter alternate_loc is used, and must be set. If this option is not used, and the location parameter is not set to a store information store number, no alternate controller support is available.

O_ALTPORT

The port of the Retail Connectivity Option TCP server started on the alternate retail controller. The optional parameter alternate_port is used, and must be set. If this option is not used, the alternate port will be defaulted to 20000. Used for TCP/IP support only.

Positional Optional Parameters

primary_port

(int) The port for the primary location. Should be between 0 and < 65535 Must be specified if O_PORT is specified.

alternate_loc (*char **) A pointer to the null-terminated name of the alternate location to use.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

This value should be set to the alternate controller if using dual controller support. Must be specified if O_ALTLOC is specified.

alternate_port (*int*) The port for the alternate location. Should be between 0 and < 65535. Must be specified if O_ALTPORT is specified.

Return Value:

- 0** psup() was successful. The remote program was started.
- 1** psup() was not successful. The errno global variable is set to indicate the error.

ERRNO Values:

- ENODEV** **3007** : The location specified was not correct, or a route to the location does not exist.
- EBADNAME** **3014** : The name or path specified is not correct.
- ENOENT** **3025** : The source 4690 file does not exist. For HCP, it is possible that the name specified could not be converted to an HCP logical name.
- EBUSY** **3029** : The 4690 resource is busy. Some other application is using the file.

EACCES

ECONNREFUSED

EHOSTUNREACH

ETIMEDOUT

ENOMEM

ELOCKED

3401 : The 4690 server can not access the file.

3425 : A connection to the remote location was refused by the remote location. Check the status of the server on the remote system.

3429 : A route to the remote host is not available. For LU0(*RTL) make sure the configuration objects are in the proper state.

3447 : The remote location did not respond within the timeout period. Check the device status. Check the remote host server.

3460 : No memory is available on the 4690 or no memory available on the AS/400.

3506 : The 4690 server can not access the file.

Usage Notes:

See the "popen - Open Retail Application File" on page 3-12 for information regarding resource authorization and locking.

For HCP Logical program name conversion for RCO servers, the QZCFAPPL is converted to the 037 CCSID and imbedded into the RCO program name. This program name is converted to the CCSID specified on the retail_ccsid for use by RCO.

For HCP Logical program name conversion for the HCP server the path is converted using the Default Job CCSID. The HCP server must be configured correctly to recognize the HCP logical.

pping - Verify Retail Application Connection

Syntax

```
#include "QRAIS/h/qzcfpcmn"
int pping(char location ,int flag, ...);
```

The pping() function is exported as qzcfping() by the *SRVPGM QZCFPCMN.

The pping() function will verify the connection to the Retail Application.

Parameters:

Required Parameters

location (input) A pointer to the null-terminated location of the retail application files.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

This can be one of the following:

- TCP/IP Host name
- Retail(LU0) Remote Location name
- Local Integrated File System ('/')
- Retail Store Information Store Number

Note: The store number is defined on the Add Retail Store Information screen. This Configuration includes the Primary Location and Secondary Location for a Dual Controller Store. This must be numeric.

flag (input) optional parameter flags.

ex:
ping_flag = O_PORT
ping_flag = O_ALTLOC | O_ALTPORT

O_PORT The port of the Retail Connectivity Option TCP server started on the primary retail controller. The optional parameter primary_port is used, and must be set. If this option is not used, the primary port will be defaulted to 20000. Used for TCP/IP support only.

O_ALTLOC The location of the Retail Connectivity Option server started on

the alternate retail controller. The optional parameter alternate_loc is used, and must be set. If this option is not used, and the location parameter is not set to a store information store number, no alternate controller support is available.

O_ALTPORT

The port of the Retail Connectivity Option TCP server started on the alternate retail controller. The optional parameter alternate_port is used, and must be set. If this option is not used, the alternate port will be defaulted to 20000. Used for TCP/IP support only.

Positional Optional Parameters

primary_port (*int*) The port for the primary location. Should be between 0 and < 65535 Must be specified if O_PORT is specified.

alternate_loc (*char **) A pointer to the null-terminated name of the alternate location to use.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

This value should be set to the alternate controller if using dual controller support. Must be specified if O_ALTLOC is specified.

alternate_port (*int*) The port for the alternate location Should be between 0 and < 65535 Must be specified if O_ALTPORT is specified.

Return Value:

- 0** pping() was successful. The location was detected.
- 1** pping() was not successful. The errno global variable is set to indicate the error.

ERRNO Values:

ENODEV **3007** : The location specified was not correct, or a route to the location does not exist.

ECONNREFUSED

3425 : A connection to the remote location was refused by the remote location. Check the status of the server on the remote system.

EHOSTUNREACH

3429 : A route to the remote host is not available. For LU0(*RTL) make sure the configuration objects are in the proper state.

ETIMEDOUT

3447 : The remote location did not respond within the timeout period. Check the

device status. Check the remote host server.

ENOMEM

3460 : No memory is available on the 4690 or no memory available on the AS/400.

Usage Notes:

See the "popen - Open Retail Application File" on page 3-12 for information regarding resource authorization and locking.

pgetprotocol - Get Retail Application File Protocol

Syntax

```
#include "QRAIS/h/qzcfpcmn"  
short pgetprotocol(char location);
```

The pgetprotocol() function is exported as qzcfgetprotocol() by the *SRVPGM QZCFPCMN.

The pgetprotocol() function will return the protocol for some specified location. If the pgetprotocol() is used for an application and you wish to have the protocol set for the whole application, you must update the CF1PROTO environment variable using psetenv() function with the returned pgetprotocol() protocol.

The checks for the location are done in following order:

1. Local IFS location check for '/' (return ZCFLOCAL)
2. gethostbyname = ipaddr (return ZCFIP)
3. Retrieve Device Description category = *RTL (return ZCFSNA)
4. return ZCFNOTFND

Parameters:

Required Parameters

location (input) A pointer to the null-terminated location of the retail application controller.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

This can be one of the following:

- TCP/IP Host name
- Retail(LU0) Remote Location name
- Local Integrated File System ('/')
- Retail Store Information Store Number

Note: The store number is defined on the Add Retail Store Information screen. This Configuration

includes the Primary Location and Secondary Location for a Dual Controller Store. This must be numeric. The protocol of the primary store location will only be checked.

Return Value:

value

If successful The protocol will be returned for the location specified. The possible protocol values defined in qzcfpcmn.h are:

ZCFSNA (0)

The location specified is an HCP (LU0) location.

ZCFIP (2)

The location specified is a TCP location.

ZCFLOCAL (3)

The location specified is a local IFS location.

value

If not successful either ZCFERROR (-3) or ZCFNOTFND (-2) will be returned. The errno global variable is set to indicate the error.

ERRNO Values:

ENODEV

3007 : The location specified was not correct, or a route to the location does not exist.

ETERM

3464 : A system error occurred.

Example

```
#include <stdlib.h>  
#include "QRAIS/h/qzcfpcmn"  
char EnvironmentVariable[3 ];  
int rc;  
char location[1 ]  
int protocol;  
protocol = (int)pgetprotocol(location);  
sprintf(EnvironmentVariable, "%s%i",  
"QZCFPROTO=", protocol);  
rc = putenv(EnvironmentVariable);
```


Chapter 4. Point of Sale/2000

This chapter gives you the information needed to work with the RAIS Point of Sale/2000 option (option 1).

Using Retail File Communications

Retail file communications allows the AS/400 system to appear as a host system to a retail controller and enables the AS/400 system to communicate using one of the following protocols:

LU0 using ADCS emulation.

TCP/IP using Retail Connectivity Option commands.

Retail File Communications Function

The Retail Application Interface Suite has an interactive menu interface and a control language interface for unattended operation. The Retail Application Interface Suite user can work with retail controller files by sending the appropriate retail file communications commands.

Retail file communications functions of the Retail Application Interface Suite can be accessed in three ways:

Through the provided AS/400 menus. Select option 1 (Work with retail communications) from the RAISCTL menu to access the retail communications commands.

Through AS/400 commands. The control language (CL) commands provided by the Retail Application Interface Suite can be performed by a CL program or entered from a display.

From an application program. An ADCS emulation high-level language application program interface (API) can be called from a language such as ILE* COBOL/400. A RAIS high-level language application program interface (API) can be called from a language such as ILE* C/400. These APIs are described in detail in Chapter 3, "Retail File Application Programming Interfaces."

Each of these methods is discussed in this chapter.

AS/400 Retail Application Interface Suite (RAISCTL) Menu

To display the Retail Application Interface Suite (RAISCTL) menu, type

```
GO QRAIS/RAISCTL
```

and then press the Enter key.

Select option 1 (Work with retail communications) from the RAISCTL menu or use the Work with Retail Communications (WRKRTLGMN) command to work with retail communications.

```
RAISCTL          AS/4  Retail Application Interface Suite
Select one of the following:

1. Work with retail communications
2. Work with retail file conversion
3. Configure retail communications application
```

Option 1 (Work with retail communications):

This option displays the Work with Retail Communications display to allow you to call the various Retail Application Interface Suite functions. Displays and information relating to this option are located on page 4-1.

Option 2 (Work with retail file conversion):

This option displays the Work with Retail File Conversion menu to allow you to work with the various conversion functions. Displays and information relating to this option are located in "Retail File Conversion."

Option 3 (Configure retail communications application):

This option displays the Configure Retail Communications Applications menu to allow you to perform configuration and maintenance functions. Displays and information relating to this option are located in Chapter 2, "Getting Started with the Retail Application Interface Suite Base Option."

Work with Retail Communications

This display provides access to the Retail Application Interface Suite functions. You can perform retail file communications functions by selecting options on this display. The following features are provided:

All store configuration entries from the retail store information file on the AS/400 system are shown.

A message stating that the store information is missing appears in the store description for controllers without a corresponding store information record.

The selected communications command is prompted with the appropriate configuration name already selected.

```
Work with Retail Communications

Position to . . . _____ Starting characters of HCP location

Type options (and remote locations), press Enter.
1=Create file   2=Send file   3=Receive file   4=Delete file
5=Work with configuration status 6=Clear file   7=Submit program...

Store
Option  Number  Location
-         -      -
-         2323  HCPC1

F3=Exit   F5=Refresh   F12=Cancel   F14=Work with store information
F23=More options
```

The following retail file communications functions can be performed from this display:

Option 1 (Create file):

This option specifies that you want to create a keyed or sequential file on a retail controller.

Option 2 (Send file):

This option specifies that you want to send a file from the AS/400 system to a retail controller. You can send the entire file or part of it.

Option 3 (Receive file):

This option specifies that you want to receive a file from a retail controller to the AS/400 system. You can receive the entire file or part of it.

Option 4 (Delete file):

This option specifies that you want to delete a file on a retail controller.

Option 5 (Work with configuration status):

This option shows a display of the status for the remote location you specify. The command issued is `WRKCFGSTS DEV CFGD(LOC) RMTLOCNAME(retail-location), where retail-location is the name of the remote-location description on the AS/400 system associated with the store information entry name selected. For the TCP/IP protocol the WRKTCPSTS OPTION(IFC) command is issued.`

Option 6 (Clear file):

This option specifies that you clear or reset a keyed or sequential file on a retail controller.

Note: For 4690 keyed files, this clears the file data only. The performance statistics in the header sector are not reset. If the performance statistics must also be reset, you should load the file with an empty file by using option 2 (Send a file to a retail controller) instead.

Option 7 (Submit program):

This option specifies that you want to start a program running on a retail controller. See the *4690 Store System: Programming Guide* for more information on the SUP (start user program) function.

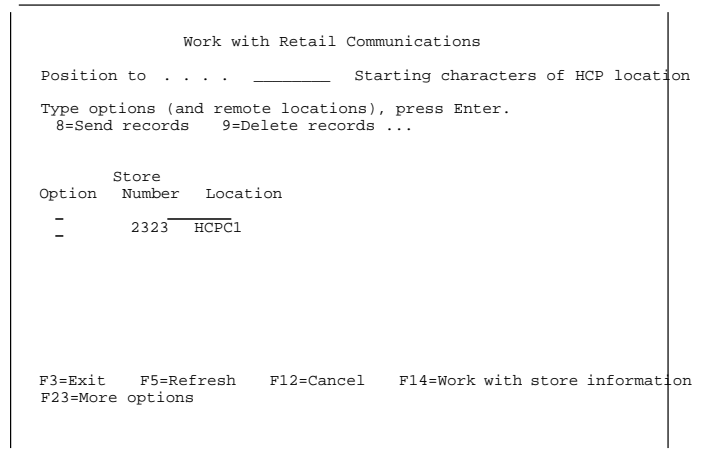
F14 (Work with store information):

This function key shows the Work with Retail Store Information display to allow the user to perform maintenance functions on the retail store information file.

F23 (More options)

Select F23 (More options) from the Work with Retail Communications menu to access this display.

This display is an extension of the previous display, Work with Retail Communications. It shows the record level commands while the prior display shows the file level commands. These options can be entered on either display.



Option 8 (Send records):

This option specifies that you want to send a file of data records to be added to or replaced in to an existing keyed file on a retail controller.

Option 9 (Delete records):

This option specifies that you want to delete a file of keyed record keys from an existing keyed file on a retail controller.

ADCS Emulation Point-of-Sale File Types Supported

The file name sent to HCP from ADCS determines both the specific point-of-sale file to be worked with and the translation to be performed. HCP is capable of exchanging four file types with the host: PC format, print format, old PSS format, and new host (new PSS) format.

PC Format Files: Personal computer (PC) format files are disk image files. No conversion is provided by HCP during transmission. Any point-of-sale file can be sent or received by point-of-sale communications in this format. Partial file dumps and loads are only valid in PC format. The retail file conversion system is designed to translate PC format files into usable AS/400 files. This is the most common retail file format exchanged with the host system.

Print Format Files: Print format files exchange reports with the host. Each print line has a leading 1-byte binary length indicator, followed by up to 255 characters of string data. HCP performs a simple ASCII to(or from) EBCDIC conversion when sending or receiving this type file to or from the host system. Any unsigned packed or binary fields are not valid in this format.

PSS Format Files: Programmable store system (PSS) format files have all the fields converted to EBCDIC in a format usable by the host. This conversion is provided by the point-of-sale application code, not the store operating system. Therefore, this format is only valid for specific named application files.

New Host (New PSS) Format Files: New host (new PSS) format files exchange point-of-sale application files with the host system. These format files have all the fields converted to EBCDIC in a format usable by the host system. This conversion is provided by the point-of-sale application, not the store operating system. Therefore, this format is only valid for specific, named application files. The converted output from PSS files has variable length fields with commas between them and leading zero suppression.

Performing this conversion in the retail controller can double the actual time required to transmit the point-of-sale data. It is recommended that larger point-of-sale networks do not use this conversion.

Retail Connectivity Option Point-of-Sale File Types Supported

The Retail Connectivity Option communications support all files as PC format files only. The retail file conversion system is designed to translate PC format files into usable AS/400 files. For Print format files, the RAIS API can be used to read and translate the ASCII formatted files to EBCDIC. See Chapter 3, "Retail File Application Programming Interfaces" on how to use the O_TEXTDATA flag when opening a file.

Retail File Communications Control Language Support

These AS/400 commands can be issued directly from an AS/400 display, called from a control language (CL) program, or used as an AS/400 command.

This table shows the AS/400 CL commands that are provided with the Retail Application Interface Suite.

Table 4-1. Retail File Communication Commands

AS/400 CL Commands	Command Description
RCVRTLF	Receive retail file from retail controller
SNDRTLF	Send retail file to retail controller
CRTRTLF	Create retail file on retail controller
DLTRTLF	Delete file on retail controller
CLRRTLF	Clear retail controller file
SNDRTLRC	Add or replace records in a keyed retail controller file
DLTRTLRC	Delete record from a keyed retail controller file
SBMRTLPGM	Start user program on retail controller

A syntax diagram follows for each of the commands. The diagrams show required and optional (defaulted) parameters for each command.

Receive Retail File (RCVRTLF) Command

The Receive Retail File (RCVRTLF) command receives a file from a retail controller. You can receive the entire file or part of it. The partial file received is only valid for files received in retail disk image format

The RCVRTLF command copies sector (data block) images of a file from the retail controller to an AS/400 database file. An entire file or a partial file can be copied based on the number of data blocks indicated on the NBRDTABLK parameter of the RCVRTLF command.

Note: ADCS data blocks are 256 bytes, although 4690 and 4684 sectors are 512 bytes. Therefore, the number of data blocks specified for a 4690 or 4684 file must be 2 times the number of sectors required.

```

RCVRTLFRTLFILE(    retail-controller-file-name    )
                  RTLLGLF
                  RTLSTMF

RMTLOCNAME(remote-location-name SEARCH          LIBL/
          IP          SNA          TOFILE(          file-name ) (P)
                  CURLIB/
                  library-name/

TOMBR(    STORE          NONE          RTLLGLF( (1) retail-logical-file )
        TOFILE          ADD          )
        member-name          REPLACE

RTLSTMF( (2) retail-stream-file )          NO
          CRTFILE(          YES          )

STRDTABLK(    FIRST          ALL
          starting-data-block-number NBRDTABLK(          number-of-data-blocks )

TCPPORT(    2
          remote-port          )
  
```

Notes:

- ¹ This parameter is valid only if *RTLLGLF is specified on the RTLFILE parameter.
- ² This parameter is valid only if *RTLSTMF is specified on the RTLFILE parameter.
- ^P All parameters preceding this point can be specified positionally.

RTLFILE Parameter

This value specifies the name of the retail controller file to be used on the retail controller.

retail-controller-file

The six-character HCP logical name of the retail controller file.

Refer to the documentation for your retail controller for a discussion on HCP logical names. Be sure you understand what format you want your file in (new host, disk image, PSS, print) and what subdirectory it resides in.

HCP on the retail controller is sensitive to uppercase and lowercase letters. Most file names must be typed in uppercase. For example, to receive the GSA Item Master file in HOST format, the retail file name entered would be +MRD 1.

This name must be entered as a quoted character string.

If the second element of the RMTLOCNAME parameter is *IP, the HCP logical name specified in this parameter is converted to the directory/filename.ext file or the application logical file or the user logical file used by the RCO server on the controller. If the conversion fails, message POS0056 is sent with a return code of 3014.

***RTLLGLF**

The retail controller file is an application logical file name or user logical file name as specified on the

retail controller. The retail logical file (RTLLGLF) parameter is used to specify the actual path name.

***RTLSTMF**

The retail controller file is a 4690 path name in the format: directory/filename.ext. The retail stream file (RTLSTMF) parameter is used to specify the actual path name.

RMTLOCNAME Parameter

Specifies the name of the remote location associated with the retail controller, and the type of connection used to connect to the retail controller.

Element 1: Name

remote-location-name: Specify the name of the remote location that is associated with the retail controller. The first element of this parameter can take several forms:

Retail remote location name. Specify a maximum of 8 characters for the remote location name. The retail remote location name is the value of the remote location name specified for the retail device description associated with the HCP session. The remote location name is specified on the RMTLOCNAME parameter on the Create Device Description (Retail) CRTDEVRTL command. The HCP session retail device description is the device description with a value of X'01' specified for the local location address. Local location address is specified on the LOCADR parameter on the CRTDEVRTL command. For example, if the retail

device description is created with RMTLOCNAME(POSC1HCP) and LOCADR(01), specify POSC1HCP for the first element of this parameter.

TCP/IP host name Specify a host name of up to 255 characters in length. The host name can be either the short form or the full domain version of the name, which is also known as the fully-qualified host name.

The host and domain names are the same as were specified on the Add TCP/IP Host Table Entry (ADDTCPHTE) and Change TCP/IP Domain (CHGTCPDMN) commands.

If this form is used, the second element of this parameter must be specified as *IP or *SEARCH.

Element 2: Type

***SEARCH:** The command determines the remote location type based upon the remote location. The utility searches for a valid remote location in the following order:

1. TCP/IP host name. If a TCP/IP host name is found that matches the value of the first element of this parameter, the utility operates as if *IP had been specified for the second element.
2. Retail remote location name. If a retail remote location name is found that matches the value of the first element of this parameter, the utility operates as if *SNA had been specified for the second element.

If both a TCP/IP host name and a retail remote location name match the name specified in the first element of this parameter, the TCP/IP host name is used, and the utility operates as if *IP had been specified for the second element.

***SNA:** The remote location has a Systems Network Architecture (SNA) address type. The utility uses the name specified in the first element as a retail remote location name.

***IP:** The remote location has an Internet Protocol (IP) address type. The utility uses the name specified in the first element as a TCP/IP host name.

TOFILE Parameter

This specifies the name of the AS/400 retail format file in which the retail controller file is to be placed. If the value entered for the TOFILE parameter does not exist, it will be created in the specified library or in *CURLIB if CRTFILE = *YES.

file-name

Specifies the file name where the file received from the retail controller will be placed.

This is the qualified name of the physical file (libraryname/filename) that receives the file. If no library qualifier is given, *LIBL is used to find the file.

*LIBL

The library list is used to locate the file name.

*CURLIB

The current library for the job is used to locate the file name. If no current library exists in the library list, QGPL is used.

library-name

Specifies the library name where files received from the retail controller will be placed.

If the TOFILE parameter does not exist before the receive (RCVRTLF) operation, the physical file is created for the TOFILE if the CRTFILE parameter is *YES. The file that is created will be a program-described file of 256-byte records. When the physical TOFILE is created, it is placed in the library specified. The creation date of the file is set to the current system date.

TOMBR Parameter

This parameter specifies the member name given to the file received from the retail controller.

The possible member values are:

*STORE

The system generates the file name in the following format:

RXXXXYYJJJ: where

R = Retail

XXXX = Store number specified for the controller during configuration

YYJJJ = Julian date that the file is received

The *STORE option is used by the system to define the name of a member within the file to be received. By the user specifying *STORE, the system knows the name of the member to be received is of the format RXXXXYYJJJ. The store number is obtained through the store information file. The store number is related to a particular location name. The store number is a 6 character field in the store information file, so only the last 4 numerical characters will be used. The date used is the job date, obtained through the RTVJOBA command.

*TOFILE

Specifies that the member will be named the same as the TOFILE name and will be the only member in the file.

member-name

Specifies the name of the member to be received from the retail controller.

If the TOMBR parameter does not exist before the receive operation, then a member will be added to the physical file TOFILE. If the member does exist, then the MBROPT(*REPLACE) can be specified to allow the receive operation to overlay the existing member. Existing records are cleared from the TOFILE member before the new records are added to the member. The

creation date of the member is set to the current system date.

MBROPT Parameter

This parameter specifies that members are created, added to, or replaced in the existing physical file members.

***NONE**

The specified TOFILE and TOMBR members are created. Members are AS/400 database files with 256-byte records. If the TOFILE does not exist, the file is created if CRTFILE=*YES. If the TOFILE does exist and TOMBR does not, a new member is created and added to the existing TOFILE.

***ADD**

The records received are added to the existing TOMBR, or if TOMBR does not exist it is created and records are added to the new member. If TOFILE and TOMBR do not exist and CRTFILE=*YES, TOFILE and TOMBR are created.

***REPLACE**

The records received replace all existing records in TOMBR. Existing records are cleared from TOMBR before the new records are added. If TOFILE and TOMBR do not exist and CRTFILE=*YES, TOFILE and TOMBR are created.

CRTFILE Parameter

This parameter specifies whether the TOFILE is created or not.

***NO** The specified TOFILE is not created.

***YES** The specified TOFILE is created if it does not exist.

RTLGLF Parameter

This value specifies the application logical file name or the user logical file name as specified on the retail controller. A maximum of 8 characters may be specified.

This parameter must be specified if *RTLGLF is specified for the RTLFILE parameter.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

RTLSTMF Parameter

This value specifies the path name to the retail controller file.

This parameter must be specified if *RTLSTMF is specified for the RTLFILE parameter.

The format of the path/file should be: *drive:/dir/file.ext* If the drive: is missing C: is the default.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

STRDTABLK Parameter

This parameter specifies the data block (sector) of the retail controller file with which you want the receive operation to start. The default is zero, specifying that the receive operation is to start at the beginning of the file. In this case, data blocks are 256 bytes, although the sector size for 4690 and 4684 files are 512 bytes. You must multiply the starting data block less one by 2 for 4690 or 4684 files. For example, if you want the receive operation to start on the fifth data block on the retail controller, you must specify a starting data block of 10. Specifying an odd-numbered data block for a 4690 or 4684 file causes the receive operation to start on the second half of data block.

***FIRST**

Specifies that the file being received from the retail controller will start at data block zero on the retail controller.

starting-data-block-number

The number of the data block on the retail file on which the receive operation starts.

NBRDTABLK Parameter

This parameter specifies the number of data blocks to be received on the AS/400 system.

***ALL**

All of the data blocks of the file are received from the retail controller.

number-of-data-blocks

Specifies the number of data blocks to be received from the retail controller.

Note: In this case data blocks are 256 bytes, although the sector size on the 4690 and 4684 controllers are 512 bytes. Therefore, the number of data blocks that you want to retrieve must be multiplied by 2 for 4690 or 4684 controllers.

TCPPORT Parameter

This parameter specifies the TCP/IP port number which the underlying retail file communications API should use to connect to the TCP/IP server on the retail controller.

20000 The default Retail Connectivity Option port number, 20000, is used.

remote-port-number Specify the port number of the Retail Connectivity Option server, in the range 1-65535.

Error Messages: Retail file communication return codes and communications errors are returned to the user message queue. If the command is entered interactively, messages are also displayed on the work station. The return codes are documented in Appendix E, "AS/400 System Error Codes"

RCVRTLF Command MBROPT Reference Matrix:

This matrix identifies whether the combinations of RCVRTLF command MBROPT parameters are valid or not valid.

Table 4-2. RCVRTLFL Command Reference Matrix

MBROPT	Member	File	Results
CRTFILE=*YES			
*NONE	member absent	file absent	Member and file created and records added to member.
	member present	file present	Error. Member and file already exist.
*ADD	member absent	file absent	Member and file created and records added to member.
	member present	file present	Records added to existing member.
*REPLACE	member absent	file absent	Member and file created and records added to member.
	member present	file present	Records cleared. Retail file replaces cleared records.
CRTFILE=*NO			
*ADD	member absent	file absent	Error, member and file do not exist
	member present	file present	Records added to existing member.
*REPLACE	member absent	file absent	Error, member and file do not exist.
	member present	file present	Records cleared. Retail file replaces cleared records.

Note: The parameter *NONE is not valid when CRTFILE=*NO.

The RCVRTLFL command uses the utility trailer record. See Appendix C, "Utility Trailer Record" for more information.

Send Retail File (SNDRTLFL) Command

The Send Retail File (SNDRTLFL) command loads a file from the AS/400 system to a retail controller. You can load the entire file or part of it. The partial load is only valid for retail controller disk image files (no implied HCP host translation).

The SNDRTLFL command processes the input required to send an ADCS Load command to the retail controller. The LOAD command copies data blocks from the AS/400 disk file to the retail controller file. An entire file or a partial file can be copied. The number of data blocks sent is based on the NBRDTABLK parameter of the command. Note that ADCS data blocks are 256 bytes, although 4690 and 4684 sectors are 512 bytes. Therefore, the number of ADCS data blocks specified for a 4690 or 4684 file must be multiplied by 2 (same as RCVRTLFL command). The user is responsible for the correct formatting of the data images. The file conversion system can be used to help in formatting the retail files.

```

SNDRTLFRTLFILE(  retail-controller-file-name  )
                  RTLLGLF
                  RTLSTMF

RMTLOCNAME(remote-location-name SEARCH SNA FROMFILE( LIBL/ file-name ) (P)
                  IP                                CURLIB/
                                                    library-name/

FROMMBR(  STORE FIRST ) DATE(  CURRENT PRV ) LOADTYPE(  INITIAL REPLACE )
          member-name          date

RTLLGLF( (1) retail-logical-file RTLSTMF( (2) retail-stream-file )

STRDTABLK(  FIRST starting-data-block-number NBRDTABLK(  ALL number-of-data-blocks )

TCPPOINT(  2 remote-port )
  
```

Notes:

- 1 This parameter is valid only if *RTLLGLF is specified on the RTLFILE parameter.
- 2 This parameter is valid only if *RTLSTMF is specified on the RTLFILE parameter.
- P All parameters preceding this point can be specified positionally.

RTLFILE Parameter

This value specifies the name of the retail controller file to be used on the retail controller.

retail-controller-file

The six-character HCP logical name of the retail controller file.

Refer to the documentation for your retail controller for a discussion on HCP logical names. Be sure you understand what format you want your file in (new host, disk image, PSS, print) and what subdirectory it resides in.

HCP on the retail controller is sensitive to uppercase and lowercase letters. Most file names must be typed in uppercase. For example, to receive the GSA Item Master file in HOST format, the retail file name entered would be +MRD 1.

This name must be entered as a quoted character string.

If the second element of the RMTLOCNAME parameter is *IP, the HCP logical name specified in this parameter is converted to the directory/filename.ext file or the application logical file or the user logical file used by the RCO server on the controller. If the conversion fails, message POS0056 is sent with a return code of 3014.

***RTLLGLF**

The retail controller file is an application logical file name or user logical file name as specified on the

retail controller. The retail logical file (RTLLGLF) parameter is used to specify the actual path name.

***RTLSTMF**

The retail controller file is a 4690 path name in the format: directory/filename.ext. The retail stream file (RTLSTMF) parameter is used to specify the actual path name.

RMTLOCNAME Parameter

Specifies the name of the remote location associated with the retail controller, and the type of connection used to connect to the retail controller.

Element 1: Name

remote-location-name: Specify the name of the remote location that is associated with the retail controller. The first element of this parameter can take several forms:

Retail remote location name. Specify a maximum of 8 characters for the remote location name. The retail remote location name is the value of the remote location name specified for the retail device description associated with the HCP session. The remote location name is specified on the RMTLOCNAME parameter on the Create Device Description (Retail) CRTDEVRTL command. The HCP session retail device description is the device description with a value of X'01' specified for the local location address. Local location address is specified on the LOCADR parameter on the CRTDEVRTL command. For example, if the retail device description is created with

RMTLOCNAME(POSC1HCP) and LOCADR(01), specify POSC1HCP for the first element of this parameter.

TCP/IP host name Specify a host name of up to 255 characters in length. The host name can be either the short form or the full domain version of the name, which is also known as the fully-qualified host name.

The host and domain names are the same as were specified on the Add TCP/IP Host Table Entry (ADDTCPHTE) and Change TCP/IP Domain (CHGTCPDMN) commands.

If this form is used, the second element of this parameter must be specified as *IP or *SEARCH.

Element 2: Type

***SEARCH:** The command determines the remote location type based upon the remote location. The utility searches for a valid remote location in the following order:

1. TCP/IP host name. If a TCP/IP host name is found that matches the value of the first element of this parameter, the utility operates as if *IP had been specified for the second element.
2. Retail remote location name. If a retail remote location name is found that matches the value of the first element of this parameter, the utility operates as if *SNA had been specified for the second element.

If both a TCP/IP host name and a retail remote location name match the name specified in the first element of this parameter, the TCP/IP host name is used, and the utility operates as if *IP had been specified for the second element.

***SNA:** The remote location has a Systems Network Architecture (SNA) address type. The utility uses the name specified in the first element as a retail remote location name.

***IP:** The remote location has an Internet Protocol (IP) address type. The utility uses the name specified in the first element as a TCP/IP host name.

FROMFILE Parameter

This parameter specifies the name of the AS/400 file that will contain the file to be sent to the retail controller.

file-name

Specifies the name of the retail AS/400 file that will contain the file to be sent to the retail controller.

This is the qualified name of the file (library name/file name) that is to be sent. If no library qualifier is given, *LIBL is used to find the file.

The AS/400 file must already exist.

***LIBL**

The library list is used to locate the file name.

***CURLIB**

The current library for the job is used to locate the file name. If no current library exists in the library list, QGPL is used.

library-name

Specifies the library name where files to be sent to the retail controller must exist.

FROMMBR Parameter

This parameter specifies the retail AS/400 member name to be sent to the retail controller.

The possible member values are:

***STORE**

The *STORE name for the file to be sent. Use this name if your AS/400 files use the same naming convention as the TOMBR (*STORE) option of the RCVRTLF command.

***FIRST**

Specifies that the first member in the file be sent.

member-name

Specifies the name of the member to be sent to the retail controller.

DATE Parameter

This parameter specifies the date for which you want to send retail data. This parameter only applies when *STORE is specified on FROMMBR.

***CURRENT**

The date used will be the current job date.

***PRV**

The date used will be the calendar day prior to the current job date.

date

Specifies the date to be processed. A 6-character date in the format of the system value QDATFMT can be entered here.

LOADTYPE Parameter

This parameter specifies whether or not this transmission is to be treated as an initial load or a partial replacement of the retail file.

***INITIAL**

This transmission will be treated as the initial load of the retail file. This means that the file will be cleared on the retail controller prior to transmission. The STRDTABLK parameter must be specified as *FIRST, and the NBRDTABLK parameter must be specified as *ALL.

***REPLACE**

This transmission will be treated as a partial replacement of the file on the retail controller.

RTLLGLF Parameter

This value specifies the application logical file name or the user logical file name as specified on the retail controller. A maximum of 8 characters may be specified.

This parameter must be specified if *RTLLGF is specified for the RTLFILE parameter.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

RTLSTMF Parameter

This value specifies the path name to the retail controller file.

This parameter must be specified if *RTLSTMF is specified for the RTLFILE parameter.

The format of the path/file should be: *drive:/dir/file.ext* If the drive: is missing C: is the default.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

STRDTABLK Parameter

This parameter specifies the data block on the retail controller file at which you want the load to start. The default is zero, specifying that the load is to start at the beginning of the file. If you have a 4690 or 4684 retail controller, see parameter STRDTABLK of the RCVRTLFL command.

***FIRST**

Specifies that the file being sent will start loading at the beginning of the file on the retail controller.

starting-data-block-number

The number of the starting data block on the retail file where the send is to start.

NBRDTABLK Parameter

This parameter specifies the number of data blocks to be sent to the retail controller. The default is *ALL, specifying that the entire file following the starting data block is to be sent. Data blocks are 256 bytes; see the starting data block parameter description for 4690 considerations when specifying number of data blocks.

***ALL**

Specifies that the entire retail file is to be sent to the retail controller.

number-of-data-blocks

Specifies the number of data blocks to be sent to the retail controller. If you have a 4690 or 4684 retail

controller, see the NBRDTABLK parameter of the RCVRTLFL command.

TCPPOINT Parameter

This parameter specifies the TCP/IP port number which the underlying retail file communications API should use to connect to the TCP/IP server on the retail controller.

20000 The default Retail Connectivity Option port number, 20000, is used.

remote-port-number Specify the port number of the Retail Connectivity Option server, in the range 1-65535.

Error Messages: Retail file communication return codes and communications errors are returned to the user message queue. If the command is entered interactively, messages are also displayed on the work station. The return codes are documented in Appendix E, "AS/400 System Error Codes"

See Appendix C, "Utility Trailer Record" for information on the utility trailer record.

Clear Retail File (CLRRTLFL) Command

The Clear Retail File (CLRRTLFL) command initializes a file on a retail controller.

The CLRRTLFL command processes the input required to send either an ADCS Load command or ADCS Reset End of Data (RED) command to the retail controller. The ADCS command sent is based on the type of file indicated on the RTLFLTYPE parameter of the CLRRTLFL command. The type is either keyed or sequential.

For keyed files, the ADCS Load command actually sends binary zeros down to the specified keyed file on the retail controller. The first sector (512 bytes) will not be initialized when using this command. This leaves the keyed file information in the header sector intact.

For sequential files, the ADCS RED (Reset End of Data) command is sent to the retail controller. This command resets the end of data pointer to zero, which effectively deletes all the data from the file.


```
CLRRTLFRTLFILE( retail-controller-file-name )
                RTLLGLF
                RTLSTMF

RMTLOCNAME(remote-location-name SEARCH SNA (P) )
                IP
                RTLFTYPE( KEYED
                        SEQUENTIAL )

RTLLGLF( (1) retail-logical-file RTLSTMF( (2) retail-stream-file )

TCPPOPT( 2
        remote-port )
```

Notes:

- ¹ This parameter is valid only if *RTLLGLF is specified on the RTLFILE parameter.
- ² This parameter is valid only if *RTLSTMF is specified on the RTLFILE parameter.
- ^P All parameters preceding this point can be specified positionally.

RTLFILE Parameter

This value specifies the name of the retail controller file to be used on the retail controller.

retail-controller-file

The six-character HCP logical name of the retail controller file.

Refer to the documentation for your retail controller for a discussion on HCP logical names. Be sure you understand what format you want your file in (new host, disk image, PSS, print) and what subdirectory it resides in.

HCP on the retail controller is sensitive to uppercase and lowercase letters. Most file names must be typed in uppercase. For example, to receive the GSA Item Master file in HOST format, the retail file name entered would be +MRD 1.

This name must be entered as a quoted character string.

If the second element of the RMTLOCNAME parameter is *IP, the HCP logical name specified in this parameter is converted to the directory/filename.ext file or the application logical file or the user logical file used by the RCO server on the controller. If the conversion fails, message POS0056 is sent with a return code of 3014.

***RTLLGLF**

The retail controller file is an application logical file name or user logical file name as specified on the retail controller. The retail logical file (RTLLGLF) parameter is used to specify the actual path name.

***RTLSTMF**

The retail controller file is a 4690 path name in the format: directory/filename.ext. The retail stream file

(RTLSTMF) parameter is used to specify the actual path name.

RMTLOCNAME Parameter

Specifies the name of the remote location associated with the retail controller, and the type of connection used to connect to the retail controller.

Element 1: Name

remote-location-name: Specify the name of the remote location that is associated with the retail controller. The first element of this parameter can take several forms:

Retail remote location name. Specify a maximum of 8 characters for the remote location name. The retail remote location name is the value of the remote location name specified for the retail device description associated with the HCP session. The remote location name is specified on the RMTLOCNAME parameter on the Create Device Description (Retail) CRTDEVRTL command. The HCP session retail device description is the device description with a value of X'01' specified for the local location address. Local location address is specified on the LOCADR parameter on the CRTDEVRTL command. For example, if the retail device description is created with RMTLOCNAME(POSC1HCP) and LOCADR(01), specify POSC1HCP for the first element of this parameter.

TCP/IP host name Specify a host name of up to 255 characters in length. The host name can be either the short form or the full domain version of the name, which is also known as the fully-qualified host name.

The host and domain names are the same as were specified on the Add TCP/IP Host Table Entry (ADDTCPHTE) and Change TCP/IP Domain (CHGTCPDMN) commands.

If this form is used, the second element of this parameter must be specified as *IP or *SEARCH.

Element 2: Type

***SEARCH:** The command determines the remote location type based upon the remote location. The utility searches for a valid remote location in the following order:

1. TCP/IP host name. If a TCP/IP host name is found that matches the value of the first element of this parameter, the utility operates as if *IP had been specified for the second element.
2. Retail remote location name. If a retail remote location name is found that matches the value of the first element of this parameter, the utility operates as if *SNA had been specified for the second element.

If both a TCP/IP host name and a retail remote location name match the name specified in the first element of this parameter, the TCP/IP host name is used, and the utility operates as if *IP had been specified for the second element.

***SNA:** The remote location has a Systems Network Architecture (SNA) address type. The utility uses the name specified in the first element as a retail remote location name.

***IP:** The remote location has an Internet Protocol (IP) address type. The utility uses the name specified in the first element as a TCP/IP host name.

RTLFTYPE Parameter

This parameter specifies the type of file to be cleared.

***KEYED**

Specifies a retail keyed file.

***SEQUENTIAL**

Specifies any file type other than keyed.

RTLLGLF Parameter

This value specifies the application logical file name or the user logical file name as specified on the retail controller. A maximum of 8 characters may be specified.

This parameter must be specified if *RTLLGF is specified for the RTLFILE parameter.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

RTLSTMF Parameter

This value specifies the path name to the retail controller file.

This parameter must be specified if *RTLSTMF is specified for the RTLFILE parameter.

The format of the path/file should be: *drive:/dir/file.ext* If the drive: is missing C: is the default.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

TCPPOINT Parameter

This parameter specifies the TCP/IP port number which the underlying retail file communications API should use to connect to the TCP/IP server on the retail controller.

20000 The default Retail Connectivity Option port number, 20000, is used.

remote-port-number Specify the port number of the Retail Connectivity Option server, in the range 1-65535.

Error Messages: Retail file communication return codes and communications errors are returned to the user message queue. If the command is entered interactively, messages are also displayed on the work station. The return codes are documented in Appendix E, "AS/400 System Error Codes"

Create Retail File (CRTRTLF) Command

The Create Retail File (CRTRTLF) command creates a keyed or sequential file on a retail controller.

The CRTRTLF command processes the input required to send an ADCS CREA command to the retail controller. The CREA command creates the space for a keyed or sequential file on the retail controller.

```

CRTRTLFRTLFILE(  retail-controller-file-name  )
                  RTLLGLF
                  RTLSTMF

RMTLOCNAME(remote-location-name  SEARCH  SNA  RTLF/TYPE(  KEYED  )
                  IP  SEQUENTIAL

NBRDTABLK(  number-of-sectorsP )
                  RTLLGLF(  (2) retail-logical-file  )

RTLSTMF(  (3) retail-stream-file  )
                  VARIABLE
                  RCDLEN(  record-length  )

VOLUME(  volume-value  ) MAXKEYLEN(  (1) 1  key-length-value  )

KEYRDMDIV(  (1) 1 randomizing-divisor-value  )

KEYMAXCHN(  (1) 4 chain-threshold-length-value  TCP/PORT(  2 remote-port  )
  
```

Notes:

- ¹ This parameter is valid only if *KEYED is specified on the RTLF/TYPE parameter.
- ² This parameter is valid only if *RTLLGLF is specified on the RTLFILE parameter.
- ³ This parameter is valid only if *RTLSTMF is specified on the RTLFILE parameter.
- ^P All parameters preceding this point can be specified positionally.

Note: The keywords MAXKEYLEN, KEYRDMDIV, and KEYMAXCHN are supported for keyed files only.

RTLFILE Parameter

This value specifies the name of the retail controller file to be used on the retail controller.

retail-controller-file

The six-character HCP logical name of the retail controller file.

Refer to the documentation for your retail controller for a discussion on HCP logical names. Be sure you understand what format you want your file in (new host, disk image, PSS, print) and what subdirectory it resides in.

HCP on the retail controller is sensitive to uppercase and lowercase letters. Most file names must be typed in uppercase. For example, to receive the GSA Item Master file in HOST format, the retail file name entered would be +MRD 1.

This name must be entered as a quoted character string.

If the second element of the RMTLOCNAME parameter is *IP, the HCP logical name specified in this parameter is converted to the directory/filename.ext file or the application logical file or the user logical file used by the RCO server on the controller. If the con-

version fails, message POS0056 is sent with a return code of 3014.

***RTLLGLF**

The retail controller file is an application logical file name or user logical file name as specified on the retail controller. The retail logical file (RTLLGLF) parameter is used to specify the actual path name.

***RTLSTMF**

The retail controller file is a 4690 path name in the format: directory/filename.ext. The retail stream file (RTLSTMF) parameter is used to specify the actual path name.

RMTLOCNAME Parameter

Specifies the name of the remote location associated with the retail controller, and the type of connection used to connect to the retail controller.

Element 1: Name

remote-location-name: Specify the name of the remote location that is associated with the retail controller. The first element of this parameter can take several forms:

Retail remote location name. Specify a maximum of 8 characters for the remote location name. The retail remote location name is the value of the remote location name specified for the retail device

description associated with the HCP session. The remote location name is specified on the RMTLOCNAME parameter on the Create Device Description (Retail) CRTDEVRTL command. The HCP session retail device description is the device description with a value of X'01' specified for the local location address. Local location address is specified on the LOCADR parameter on the CRTDEVRTL command. For example, if the retail device description is created with RMTLOCNAME(POSC1HCP) and LOCADR(01), specify POSC1HCP for the first element of this parameter.

TCP/IP host name Specify a host name of up to 255 characters in length. The host name can be either the short form or the full domain version of the name, which is also known as the fully-qualified host name.

The host and domain names are the same as were specified on the Add TCP/IP Host Table Entry (ADDTCPHTE) and Change TCP/IP Domain (CHGTCPDMN) commands.

If this form is used, the second element of this parameter must be specified as *IP or *SEARCH.

Element 2: Type

***SEARCH:** The command determines the remote location type based upon the remote location. The utility searches for a valid remote location in the following order:

1. TCP/IP host name. If a TCP/IP host name is found that matches the value of the first element of this parameter, the utility operates as if *IP had been specified for the second element.
2. Retail remote location name. If a retail remote location name is found that matches the value of the first element of this parameter, the utility operates as if *SNA had been specified for the second element.

If both a TCP/IP host name and a retail remote location name match the name specified in the first element of this parameter, the TCP/IP host name is used, and the utility operates as if *IP had been specified for the second element.

***SNA:** The remote location has a Systems Network Architecture (SNA) address type. The utility uses the name specified in the first element as a retail remote location name.

***IP:** The remote location has an Internet Protocol (IP) address type. The utility uses the name specified in the first element as a TCP/IP host name.

RTLFTYPE Parameter

This parameter specifies the type of file to be created. Specify `KEYED` for a keyed file. Specify `SEQUENTIAL` for any file type other than keyed.

NBRDTABLK Parameter

This parameter specifies the number of data blocks (256 bytes) to be created in the retail controller file. For 3651 files, you must specify a multiple of 12.

RTLLGLF Parameter

This value specifies the application logical file name or the user logical file name as specified on the retail controller. A maximum of 8 characters may be specified.

This parameter must be specified if *RTLLGF is specified for the RTLFILE parameter.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

RTLSTMF Parameter

This value specifies the path name to the retail controller file.

This parameter must be specified if *RTLSTMF is specified for the RTLFILE parameter.

The format of the path/file should be: *drive:/dir/file.ext* If the drive: is missing C: is the default.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

RCDLEN Parameter

This parameter specifies the record length in bytes.

For a 4690 or 4684 keyed file, you can specify from 1 to 508. For a 3651 or 3684 keyed file with fixed length records, you can specify from 1 to 254. For a 3651 or 3684 keyed file with variable length records, specify `VARIABLE`.

For a 4690 or 4684 non-keyed file, specify `VARIABLE`. For a 3651 or 3684 sequential file with fixed length records, you can specify from 1 to 256. For a 3651 or 3684 sequential file with variable length records, specify `VARIABLE`.

VOLUME Parameter

This parameter specifies the drive on which the file is to be created. For 4690 controllers or for 3651 and 3684 controllers with a single disk drive, you must specify zero. For 3651 and 3684 controllers that have multiple disks attached, you can specify from 0 to 3. The default is zero.

MAXKEYLEN Parameter

This parameter specifies the key length in bytes. You can specify any value from 1 byte to a maximum of the record length.

KEYRMDIV Parameter

This parameter specifies the randomizing divisor for a keyed file. You should specify an odd number that is less than the number of sectors in the file. For optimal use of the file, this number should not be divisible by 3 or 5.

KEYMAXCHN Parameter

This parameter specifies the acceptable length of the overflow chains for the file to be created. When adding records to the file, the system logs a message in the message log if an overflow chain is encountered that exceeds the length of the chaining threshold. The message indicates either that the keyed file is getting too full or that due to the randomizing divisor value specified, the file was not randomized effectively. The default value is 4, which is also the recommended value. For more detailed information on the KEYMAXCHN parameter, see the *IBM 4690 Store System: Programmer's Guide*.

TCPPORT Parameter

This parameter specifies the TCP/IP port number which the underlying retail file communications API should use to connect to the TCP/IP server on the retail controller.

20000 The default Retail Connectivity Option port number, 20000, is used.

remote-port-number Specify the port number of the Retail Connectivity Option server, in the range 1-65535.

For HCP configurations this command only creates the space on the retail controller. If you want to create a keyed file, you must follow this command with one of the following options. Use the SBMRTLPGM command to start the keyed file utility on the 4690. This will create the 512-byte header record in the keyed file. Or use the SNDRTLF command to load a formatted keyed file onto the retail controller.

Note: See Appendix D, "QRTXHASH Randomizing Routine" for instructions or you can create a direct file of the data to be loaded. Send that file to the retail controller and then use the keyed file utility to create a keyed file from the direct file.

Error Messages: Retail file communication return codes and communications errors are returned to the user message queue. If the command is entered interactively, messages are also displayed on the work station. The return codes are documented in Appendix E, "AS/400 System Error Codes"

Delete Retail File (DLTRTLF) Command

The Delete Retail File (DLTRTLF) command deletes a file from a retail controller.

The DLTRTLF command processes the input required to send an ADCS PURG command to the retail controller. The PURG command deletes a file from the retail controller.

```

Job: B,I Pgm: B,I REXX: B,I Exec
Threadsaf: No

DLTRTLFRTLFILE( retail-controller-file-name )
                RTLLGLF
                RTLSTMF

RMTLOCNAME(remote-location-name SEARCH SNA (P) )
                IP RTLLGLF( (1) retail-logical-file )

RTLSTMF( (2) retail-stream-file )
                TCPPOINT( remote-port )

```

Notes:

- ¹ This parameter is valid only if *RTLLGLF is specified on the RTLFILE parameter.
- ² This parameter is valid only if *RTLSTMF is specified on the RTLFILE parameter.

^P All parameters preceding this point can be specified positionally.

RTLFILE Parameter

This value specifies the name of the retail controller file to be used on the retail controller.

retail-controller-file

The six-character HCP logical name of the retail controller file.

Refer to the documentation for your retail controller for a discussion on HCP logical names. Be sure you understand what format you want your file in (new

host, disk image, PSS, print) and what subdirectory it resides in.

HCP on the retail controller is sensitive to uppercase and lowercase letters. Most file names must be typed in uppercase. For example, to receive the GSA Item Master file in HOST format, the retail file name entered would be +MRD 1.

This name must be entered as a quoted character string.

If the second element of the RMTLOCNAME parameter is *IP, the HCP logical name specified in this parameter is converted to the directory/filename.ext file or the application logical file or the user logical file used by the RCO server on the controller. If the conversion fails, message POS0056 is sent with a return code of 3014.

***RTLLGLF**

The retail controller file is an application logical file name or user logical file name as specified on the retail controller. The retail logical file (RTLLGLF) parameter is used to specify the actual path name.

***RTLSTMF**

The retail controller file is a 4690 path name in the format: directory/filename.ext. The retail stream file (RTLSTMF) parameter is used to specify the actual path name.

RMTLOCNAME Parameter

Specifies the name of the remote location associated with the retail controller, and the type of connection used to connect to the retail controller.

Element 1: Name

remote-location-name: Specify the name of the remote location that is associated with the retail controller. The first element of this parameter can take several forms:

Retail remote location name. Specify a maximum of 8 characters for the remote location name. The retail remote location name is the value of the remote location name specified for the retail device description associated with the HCP session. The remote location name is specified on the RMTLOCNAME parameter on the Create Device Description (Retail) CRTDEVRTL command. The HCP session retail device description is the device description with a value of X'01' specified for the local location address. Local location address is specified on the LOCADR parameter on the CRTDEVRTL command. For example, if the retail device description is created with RMTLOCNAME(POSC1HCP) and LOCADR(01), specify POSC1HCP for the first element of this parameter.

TCP/IP host name Specify a host name of up to 255 characters in length. The host name can be either the short form or the full domain version of the name, which is also known as the fully-qualified host name.

The host and domain names are the same as were specified on the Add TCP/IP Host Table Entry (ADDTCPHTE) and Change TCP/IP Domain (CHGTCPDMN) commands.

If this form is used, the second element of this parameter must be specified as *IP or *SEARCH.

Element 2: Type

***SEARCH:** The command determines the remote location type based upon the remote location. The utility searches for a valid remote location in the following order:

1. TCP/IP host name. If a TCP/IP host name is found that matches the value of the first element of this parameter, the utility operates as if *IP had been specified for the second element.
2. Retail remote location name. If a retail remote location name is found that matches the value of the first element of this parameter, the utility operates as if *SNA had been specified for the second element.

If both a TCP/IP host name and a retail remote location name match the name specified in the first element of this parameter, the TCP/IP host name is used, and the utility operates as if *IP had been specified for the second element.

***SNA:** The remote location has a Systems Network Architecture (SNA) address type. The utility uses the name specified in the first element as a retail remote location name.

***IP:** The remote location has an Internet Protocol (IP) address type. The utility uses the name specified in the first element as a TCP/IP host name.

RTLLGLF Parameter

This value specifies the application logical file name or the user logical file name as specified on the retail controller. A maximum of 8 characters may be specified.

This parameter must be specified if *RTLLGF is specified for the RTLFILE parameter.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

RTLSTMF Parameter

This value specifies the path name to the retail controller file.

This parameter must be specified if *RTLSTMF is specified for the RTLFILE parameter.

The format of the path/file should be: *drive:/dir/file.ext* If the drive: is missing C: is the default.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

TCPPORT Parameter

This parameter specifies the TCP/IP port number which the underlying retail file communications API should use to connect to the TCP/IP server on the retail controller.

20000 The default Retail Connectivity Option port number, 20000, is used.

remote-port-number Specify the port number of the Retail Connectivity Option server, in the range 1-65535.

Error Messages: Retail file communication return codes and communications errors are returned to the user message queue. If the command is entered interactively, messages

are also displayed on the work station. The return codes are documented in Appendix E, "AS/400 System Error Codes"

Send Retail Record (SNDRTLRCDD) Command

The Send Retail Record (SNDRTLRCDD) command allows you to complete a ADCS KEYD add or replace record command. This command allows you to add or change individual keyed records on the retail controller. The retail controller limits the record size supported by this function to 256 bytes. The user provides an input file (REFFILE or REFMBR parameters) of individual records to be added or changed. The RPLRCDOPT parameter specifies how the record will be updated.

```
Job: B,I Pgm: B,I REXX: B,I Exec
Threadsaf: No

SNDRTLRCDD( retail-controller-file-name )
           RTLLGLF
           RTLSTMF

RMTLOCNAME( remote-location-name SEARCH SNA REFFILE( LIBL/ reference-file-name )
           IP                                CURLIB/
                                           library-name/

RCDLEN( record-length (P) )

REFMBR( FIRST member-name ) RPLRCDOPT( BOTH REPLACE )
                                           ADD

RTLLGLF( (1) retail-logical-file RTLSTMF( (2) retail-stream-file )

TCPPOPT( DFT remote-port )

Notes:
1 This parameter is valid only if *RTLLGLF is specified on the RTLFILE parameter.
2 This parameter is valid only if *RTLSTMF is specified on the RTLFILE parameter.
P All parameters preceding this point can be specified positionally.
```

RTLFILE Parameter

This value specifies the name of the retail controller file to be used on the retail controller.

retail-controller-file

The six-character HCP logical name of the retail controller file.

Refer to the documentation for your retail controller for a discussion on HCP logical names. Be sure you understand what format you want your file in (new host, disk image, PSS, print) and what subdirectory it resides in.

HCP on the retail controller is sensitive to uppercase and lowercase letters. Most file names must be typed in uppercase. For example, to receive the GSA Item Master file in HOST format, the retail file name entered would be +MRD 1.

This name must be entered as a quoted character string.

If the second element of the RMTLOCNAME parameter is *IP, the HCP logical name specified in this parameter is converted to the directory/filename.ext file or the application logical file or the user logical file used by the RCO server on the controller. If the conversion fails, message POS0056 is sent with a return code of 3014.

*RTLLGLF

The retail controller file is an application logical file name or user logical file name as specified on the retail controller. The retail logical file (RTLLGLF) parameter is used to specify the actual path name.

*RTLSTMF

The retail controller file is a 4690 path name in the format: directory/filename.ext. The retail stream file

(RTLSTMF) parameter is used to specify the actual path name.

RMTLOCNAME Parameter

Specifies the name of the remote location associated with the retail controller, and the type of connection used to connect to the retail controller.

Element 1: Name

remote-location-name: Specify the name of the remote location that is associated with the retail controller. The first element of this parameter can take several forms:

Retail remote location name. Specify a maximum of 8 characters for the remote location name. The retail remote location name is the value of the remote location name specified for the retail device description associated with the HCP session. The remote location name is specified on the RMTLOCNAME parameter on the Create Device Description (Retail) CRTDEVRTL command. The HCP session retail device description is the device description with a value of X'01' specified for the local location address. Local location address is specified on the LOCADR parameter on the CRTDEVRTL command. For example, if the retail device description is created with RMTLOCNAME(POSC1HCP) and LOCADR(01), specify POSC1HCP for the first element of this parameter.

TCP/IP host name Specify a host name of up to 255 characters in length. The host name can be either the short form or the full domain version of the name, which is also known as the fully-qualified host name.

The host and domain names are the same as were specified on the Add TCP/IP Host Table Entry (ADDTCPHTE) and Change TCP/IP Domain (CHGTCPDMN) commands.

If this form is used, the second element of this parameter must be specified as *IP or *SEARCH.

Element 2: Type

***SEARCH:** The command determines the remote location type based upon the remote location. The utility searches for a valid remote location in the following order:

1. TCP/IP host name. If a TCP/IP host name is found that matches the value of the first element of this parameter, the utility operates as if *IP had been specified for the second element.
2. Retail remote location name. If a retail remote location name is found that matches the value of the first element of this parameter, the utility operates as if *SNA had been specified for the second element.

If both a TCP/IP host name and a retail remote location name match the name specified in the first element of

this parameter, the TCP/IP host name is used, and the utility operates as if *IP had been specified for the second element.

***SNA:** The remote location has a Systems Network Architecture (SNA) address type. The utility uses the name specified in the first element as a retail remote location name.

***IP:** The remote location has an Internet Protocol (IP) address type. The utility uses the name specified in the first element as a TCP/IP host name.

REFFILE Parameter

This parameter specifies the name of the AS/400 file that contains the REFMBR for the retail controller.

reference-file-name

Specifies the name of the reference file.

*LIBL

The library list is used to locate the file name.

*CURLIB

The current library for the job is used to locate the file name. If no current library exists in the library list, QGPL is used.

library-name

Specifies the library where the source files are kept.

RCDLEN Parameter

This parameter is the record length in bytes of the AS/400 data that you want to send. If the AS/400 data is in host format, be sure to include the carriage return line feed characters (X'0D0A ') in the length calculation. Valid values are from 0 to 256. HCP does not support record lengths greater than 256 bytes.

REFMBR Parameter

This parameter specifies the member name of the specified REFFILE and library. This file contains the records that you want to add or replace on the retail controller file. Any number of records can be placed in this file.

Reference files are AS/400 database files with a sequential organization. When you create the reference file, you should specify a record length that is long enough to hold a record for the file to be accessed. Retail controller files with record lengths greater than 256 bytes are not accessible using the SNDRTLRC command.

The possible member values are:

*FIRST

This value specifies that the first member in the file will be sent.

member-name

This value specifies the name of the member to be sent to the retail controller.

RPLRCDOPT Parameter

This parameter specifies how the record update on the retail controller will be handled.

The possible member values are:

***BOTH**

Specifies that the record will be replaced if it exists on retail controller or added if it does not exist. No error messages are sent to the AS/400 system when using this option.

***REPLACE**

Specifies that the record will be replaced if it exists on the retail controller. An error message is sent back from HCP if the record does not exist.

***ADD**

This causes the record to be added to the specified retail file on the retail controller. If the record already exists, HCP will send an error message back to the AS/400 system.

RTLLGLF Parameter

This value specifies the application logical file name or the user logical file name as specified on the retail controller. A maximum of 8 characters may be specified.

This parameter must be specified if *RTLLGF is specified for the RTLFILE parameter.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

RTLSTMF Parameter

This value specifies the path name to the retail controller file.

This parameter must be specified if *RTLSTMF is specified for the RTLFILE parameter.

The format of the path/file should be: *drive:/dir/file.ext* If the drive: is missing C: is the default.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

TCPPORT Parameter

This parameter specifies the TCP/IP port number which the underlying retail file communications API should use to connect to the TCP/IP server on the retail controller.

20000 The default Retail Connectivity Option port number, 20000, is used.

remote-port-number Specify the port number of the Retail Connectivity Option server, in the range 1-65535.

Any number of ADD KEYED record accesses intended for the same retail controller file can be listed in one reference file.

Error Messages: Retail file communication return codes and communications errors are returned to the user message queue. If the command is entered interactively, messages are also displayed on the work station. The return codes are documented in Appendix E, "AS/400 System Error Codes"

Reference Member Format Example: The following is an example of an AS/400 reference member record to add or replace a record in a retail controller file. The example adds an item to the **General Sales Application (GSA) Item Master** file. The GSA is the 4690 Checkout Application designed for mass merchandisers. Records can be built on the AS/400 system for the retail controller in HOST translated format or PC format. This example adds one item in Host format. The record length refers to the length of the data on the AS/400 system to be sent. The RCDLEN parameter for this record is 53, which includes the carriage return line feed.

Data in AS/4 REFERENCE file

```
123456789, , , ,999,8,19999, ,Item Description, ,##
```

Note that this record is in HOST format and fields are delimited by commas. All records are ended with X'0D0A' for carriage return and line feed. The X'0D0A' is shown as a ## in the data example, because they are unprintable hex characters. This example maps the GSA Item Master file in the *IBM 4690 General Sales Application (GSA): Programming Guide*.

The first field is the key of the item number. Remaining fields are the indicator bytes, department number, description, sale quantity, and price. (See the *IBM 4690 GSA Programming Guide* for file layouts or the *4690 Store System: Programming Guide, Managing Your Files* for more information on files.)

Submit Retail Program (SBMRTLPGM) Command

The Submit Retail Program (SBMRTLPGM) command submits a program for batch processing on a retail controller.

The SBMRTLPGM command processes the input required to send an ADCS Start User Program (SUP) command to the retail controller.

```
SBMRTLPGMRTLPGM(    retail-controller-file-name    )  
                    RTLLGLF  
                    RTLSTMF  
  
RMTLOCNAME(remote-location-name    SEARCH  
                    SNA    (P)    )  
                    IP    RTLLGLF( (1) retail-logical-file    )  
  
RTLSTMF( (2) retail-stream-file    )  
                    TCPPORT(    2  
                    remote-port    )
```

Notes:

- 1 This parameter is valid only if *RTLLGLF is specified on the RTLFILE parameter.
- 2 This parameter is valid only if *RTLSTMF is specified on the RTLFILE parameter.
- P All parameters preceding this point can be specified positionally.

RTLPGM Parameter

This value specifies the name of the retail controller program that is to be started.

retail-controller-program

The 8 character logical name of the retail controller program. This name must be the program logical name as specified on a 4690 controller or the application definition record name as specified on a 3651 or 3684 controller.

This name must be entered as a quoted character string.

If the second element of the RMTLOCNAME parameter is *IP, an HCP logical name specified in this parameter is converted to the directory/program.ext file or the application logical file or the user logical file used by the RCO server on the controller.

***RTLLGLF**

The retail controller program is an application logical file name or user logical file name as specified on the retail controller. The retail logical file (RTLLGLF parameter) is used to specify the actual path name.

***RTLSTMF**

The retail controller program is a path name in the format: directory/program.ext. The retail stream file (RTLSTMF parameter) is used to specify the actual path name.

RMTLOCNAME Parameter

Specifies the name of the remote location associated with the retail controller, and the type of connection used to connect to the retail controller.

Element 1: Name

remote-location-name: Specify the name of the remote location that is associated with the retail controller. The first element of this parameter can take several forms:

Retail remote location name. Specify a maximum of 8 characters for the remote location name. The

retail remote location name is the value of the remote location name specified for the retail device description associated with the SUP session. The remote location name is specified on the RMTLOCNAME parameter on the Create Device Description (Retail) CRTDEVRTL command. The SUP session retail device description is the device description with a value of *SBMRTLPGM for the application type. Application type is specified on the APPTYPE parameter on the the CRTDEVRTL command. For example, if the retail device description is created with RMTLOCNAME(POSC1HCP) and LOCADR(01), specify POSC1HCP for the first element of this parameter.

TCP/IP host name Specify a host name of up to 255 characters in length. The host name can be either the short form or the full domain version of the name, which is also known as the fully-qualified host name.

The host and domain names are the same as were specified on the Add TCP/IP Host Table Entry (ADDTCPHTE) and Change TCP/IP Domain (CHGTCPDMN) commands.

If this form is used, the second element of this parameter must be specified as *IP or *SEARCH.

Element 2: Type

***SEARCH:** The command determines the remote location type based upon the remote location. The utility searches for a valid remote location in the following order:

1. TCP/IP host name. If a TCP/IP host name is found that matches the value of the first element of this parameter, the utility operates as if *IP had been specified for the second element.
2. Retail remote location name. If a retail remote location name is found that matches the value of the

first element of this parameter, the utility operates as if *SNA had been specified for the second element.

If both a TCP/IP host name and a retail remote location name match the name specified in the first element of this parameter, the TCP/IP host name is used, and the utility operates as if *IP had been specified for the second element.

***SNA:** The remote location has a Systems Network Architecture (SNA) address type. The utility uses the name specified in the first element as a retail remote location name.

***IP:** The remote location has an Internet Protocol (IP) address type. The utility uses the name specified in the first element as a TCP/IP host name.

RTLLGLF Parameter

This value specifies the application logical file name or the user logical file name as specified on the retail controller. A maximum of 8 characters may be specified.

This parameter must be specified if *RTLLGF is specified for the RTLFILE parameter.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

RTLSTMF Parameter

This value specifies the path name to the retail controller file.

This parameter must be specified if *RTLSTMF is specified for the RTLFILE parameter.

The format of the path/file should be: *drive:/dir/file.ext* If the drive: is missing C: is the default.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

TCPPORT Parameter

This parameter specifies the TCP/IP port number which the underlying retail file communications API should use to connect to the TCP/IP server on the retail controller.

20000 The default Retail Connectivity Option port number, 20000, is used.

remote-port-number Specify the port number of the Retail Connectivity Option server, in the range 1-65535.

Error Messages: Retail file communication return codes and communications errors are returned to the user message queue. If the command is entered interactively, messages are also displayed on the work station. The return codes are documented in Appendix E, "AS/400 System Error Codes"

Delete Retail Record (DLTRTLRCD) Command

The Delete Retail Record (DLTRTLRCD) command deletes a series of keyed records in a retail controller file.

The DLTRTLRCD command processes the input required to send an (ADCS) KEYD delete command to the retail controller. The REFMBR contains the records to be deleted.

```
DLTRTLRCIRTLFILE(  retail-controller-file-name  )
                   RTLLGLF
                   RTLSTMF

RMTLOCNAME(remote-location-name SEARCH SNA REF)ILE( LIBL/          reference-file-name  )
                   IP          CURLIB/
                   library-name/

KEYLEN(  key-length-value(P) )

                   FIRST
                   REFMBR(  member-name  )

RTLLGLF( (1) retail-logical-file  RTLSTMF( (2) retail-stream-file  )          NO
                   RTNERR(          YES          )

TCPPORT( 2 remote-port  )
```

Notes:

- ¹ This parameter is valid only if *RTLLGLF is specified on the RTLFILE parameter.
- ² This parameter is valid only if *RTLSTMF is specified on the RTLFILE parameter.
- ^P All parameters preceding this point can be specified positionally.

RTLFILE Parameter

This value specifies the name of the retail controller file to be used on the retail controller.

retail-controller-file

The six-character HCP logical name of the retail controller file.

Refer to the documentation for your retail controller for a discussion on HCP logical names. Be sure you understand what format you want your file in (new host, disk image, PSS, print) and what subdirectory it resides in.

HCP on the retail controller is sensitive to uppercase and lowercase letters. Most file names must be typed in uppercase. For example, to receive the GSA Item Master file in HOST format, the retail file name entered would be +MRD 1.

This name must be entered as a quoted character string.

If the second element of the RMTLOCNAME parameter is *IP, the HCP logical name specified in this parameter is converted to the directory/filename.ext file or the application logical file or the user logical file used by the RCO server on the controller. If the conversion fails, message POS0056 is sent with a return code of 3014.

***RTLLGLF**

The retail controller file is an application logical file name or user logical file name as specified on the retail controller. The retail logical file (RTLLGLF) parameter is used to specify the actual path name.

***RTLSTMF**

The retail controller file is a 4690 path name in the format: directory/filename.ext. The retail stream file (RTLSTMF) parameter is used to specify the actual path name.

RMTLOCNAME Parameter

Specifies the name of the remote location associated with the retail controller, and the type of connection used to connect to the retail controller.

Element 1: Name

remote-location-name: Specify the name of the remote location that is associated with the retail controller. The first element of this parameter can take several forms:

Retail remote location name. Specify a maximum of 8 characters for the remote location name. The retail remote location name is the value of the remote location name specified for the retail device description associated with the HCP session. The remote location name is specified on the RMTLOCNAME parameter on the Create Device Description (Retail) CRTDEVRTL command. The HCP session retail device description is the device description with a value of X'01' specified for the local location address. Local location address is specified on the LOCADR parameter on the CRTDEVRTL command. For example, if the retail device description is created with RMTLOCNAME(POSC1HCP) and LOCADR(01), specify POSC1HCP for the first element of this parameter.

TCP/IP host name Specify a host name of up to 255 characters in length. The host name can be either the short form or the full domain version of the name, which is also known as the fully-qualified host name.

The host and domain names are the same as were specified on the Add TCP/IP Host Table Entry (ADDTCPHTE) and Change TCP/IP Domain (CHGTCPDMN) commands.

If this form is used, the second element of this parameter must be specified as *IP or *SEARCH.

Element 2: Type

***SEARCH:** The command determines the remote location type based upon the remote location. The utility searches for a valid remote location in the following order:

1. TCP/IP host name. If a TCP/IP host name is found that matches the value of the first element of this parameter, the utility operates as if *IP had been specified for the second element.
2. Retail remote location name. If a retail remote location name is found that matches the value of the first element of this parameter, the utility operates as if *SNA had been specified for the second element.

If both a TCP/IP host name and a retail remote location name match the name specified in the first element of this parameter, the TCP/IP host name is used, and the utility operates as if *IP had been specified for the second element.

***SNA:** The remote location has a Systems Network Architecture (SNA) address type. The utility uses the name specified in the first element as a retail remote location name.

***IP:** The remote location has an Internet Protocol (IP) address type. The utility uses the name specified in the first element as a TCP/IP host name.

REFFILE Parameter

This parameter specifies the name of the AS/400 file that contains the REFMBR to be sent to the retail controller.

reference-file-name

Specifies the name of the AS/400 file that contains the reference member to be sent to the retail controller.

*LIBL

The library list is used to locate the file name.

*CURLIB

The current library for the job is used to locate the file name. If no current library exists in the library list, QGPL is used.

library-name

Specifies the library where the reference files are kept.

KEYLEN Parameter

This parameter specifies the length in bytes of the key to send to the retail controller.

REFMBR Parameter

This parameter specifies the member name of the specified REFFILE and Library. It will contain the key of the record to delete on the retail controller. Any number of deletes can be placed in this file. Reference files are AS/400 database files with a sequential organization. When you create the reference file, you should specify a record length that is long enough to hold a key from the file to be accessed.

*FIRST

Specifies that the first member in the file be sent to the retail controller.

member-name

Specifies the name of the member to be sent to the retail controller.

RTLLGLF Parameter

This value specifies the application logical file name or the user logical file name as specified on the retail controller. A maximum of 8 characters may be specified.

This parameter must be specified if *RTLLGF is specified for the RTLFILE parameter.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

RTLSTMF Parameter

This value specifies the path name to the retail controller file.

This parameter must be specified if *RTLSTMF is specified for the RTLFILE parameter.

The format of the path/file should be: *drive:/dir/file.ext* If the drive: is missing C: is the default.

If the second element of the RMTLOCNAME parameter is *SNA, the file name specified in this parameter is converted to a valid 6 character HCP file name. If the conversion fails, message POS0056 is sent with a return code of 3014.

RTNERR Parameter

This parameter specifies how HCP errors will be handled on the retail controller.

The possible member values are:

*NO

An error status is not returned to the AS/400 system if the record is not in the file.

*YES

An error status is returned to the AS/400 system if the record is not in the file.

TCPPORT Parameter

This parameter specifies the TCP/IP port number which the underlying retail file communications API should use to connect to the TCP/IP server on the retail controller.

20000 The default Retail Connectivity Option port number, 20000, is used.

remote-port-number Specify the port number of the Retail Connectivity Option server, in the range 1-65535.

Any number of keyed record accesses intended for the same retail controller file can be listed in one reference file.

Reference Member Format Example for Delete Record:

The following is an example of an AS/400 reference member record to delete a record in a retail controller file. The example deletes an item in the GSA (General Sales Application) Item Master file. Records can be built on the AS/400 system for the retail controller in HOST translated format or PC format. This example deletes one item in Host format. The Record Length refers to the length of the data on the AS/400 system to be sent. The RCDLEN parameter for this record is 11.

```
Data in AS/4 REFERENCE file
```

```
123456789 (key of record to delete)
```

When deleting records from keyed files other than the Item file, the relative position of the key must be correct. If the key is in position 2, then the example would look like this:

```
, 123456789.
```

The record length would be 13.

Error Messages: Retail file communication return codes and communications errors are returned to the user message queue. If the command is entered interactively, messages are also displayed on the work station. The return codes are documented in Appendix E, "AS/400 System Error Codes"

Using Host Command Processor (HCP) Emulation

HCP emulation functions are supported by the Retail Application Interface Suite for communications with the host computer ADCS program.

The HCP emulation responds to ADCS host commands as if the AS/400 system is a retail controller. The AS/400 in-store processor should fit into a host-controlled point-of-sale

network with little or no adjustment to the host communications functions.

HCP emulation requires very little setup on the AS/400 system. The AS/400 host communications line must be configured using the SNA upline facility (SNUF) and the host system must be configured to support the Retail Application Interface Suite's HCP emulation. There is no operator interaction with HCP emulation because it is controlled entirely by the host system.

HCP Commands Supported

HCP emulation does not provide any conversion capabilities while it is running. Files previously converted by the file conversion can be retrieved by the host system using the ADCS DUMP command.

The ADCS host user can issue the following commands to HCP emulation on the AS/400 system:

Dump	Retrieve a complete or partial AS/400 file.
Load	Send a complete or partial file to the AS/400 system.
Create	Create a new empty file on the AS/400 system.
Reset	Reset the end of data pointer on a sequential file.
Clear	Zero all the records in an AS/400 file.
Purge	Delete an AS/400 file.
Status	Check on the status of the previous command.

Work with Host File Cross References

Select option 3 (Work with host file cross references) from the Configure Retail Communications Application menu to access this display.

Use this display to set up and maintain the host file name cross-reference list for the host command processor (HCP) emulation.

This cross-reference list is used only by HCP emulation to map the Advanced Data Communications for Stores (ADCS) 6-character file names to the desired AS/400 file and library names. The ADCS 6-character name has special characters and in most cases will not be a valid AS/400 name. All files to be exchanged with the ADCS host system must be in this cross-reference list. A file name message that is not valid (X'3FFD') will be returned to the ADCS host for any ADCS command referring to a file not in this list. If your AS/400 system is not going to use HCP emulation to an ADCS host system, you do not need to create this host file cross-reference list.

```

Work with Host File Cross References

Position to . . . . . Starting characters of host file

Type options, press Enter.
  2=Change  4=Delete  7=Rename

Host      -----AS/4 -----
Option  File      File      Library  Host File Description
-      EAMSL5  EAMSALES  QPOSFILES  SMKT ITEM MOVEMENT CURRENT
-      EGRDAY  EAMDAILY  QPOSFILES  SMKT DAILY5 SALES FIGURES
-      DAVES1  DCPHCPTST DCPLIB   TEST FILE FOR HCP EMULATION
-      /MRD 1  PCITEMS  QPOSFILES  ITEM FILE UNTRANSLATED
-      +MRD 1  ITEMMAST  QPOSF     ITEM MASTER GSA HOST TRANSLATE

F3=Exit  F5=Refresh  F6=Add host file  F12=Cancel  F21=Print list

```

The following function keys are available on the Work with Host File Cross References menu to help in working with cross references:

F6=Add host file:

Used to create or add a new host file cross-reference entry to the host file cross-reference list.

F21=Print list:

Prints the retail host cross-reference list.

Add Host File Cross Reference

Select F6 (Add host file) from the Work with Host File Cross References menu to access this display.

This display is used to add to the host file cross-reference list. AS/400 files to be used with HCP emulation must be cross-referenced with their corresponding 6-character host name. The AS/400 files must also be single member files with 256-byte physical records.

```

Add Host File Cross Reference

Type choices, press Enter.

Host file . . . . .

AS/4 file . . . . . Name
AS/4 library . . . . . Name

Host file description . . . . .

F3=Exit  F12=Cancel

```

Change Host File Cross Reference

Select option 2 (Change) from the Work with Host File Cross References menu to access this display.

This display is used to change the AS/400 file name, library name, or text associated with a host file name.

```

Change Host File Cross Reference

Host file . . . . . : +MRD 1

Type choices, press Enter.

AS/4 file . . . . . ITEMMASTName
AS/4 library . . . . . QPOSF Name

Host file description . . . . . ITEM MASTER GSA HOST TRANSLATE

F3=Exit  F12=Cancel

```

Confirm Delete of Host File Cross References

Select option 4 (Delete) from the Work with Host File Cross References menu to access this display.

This display is used to confirm a deletion from the host file cross-reference list.

```

Confirm Delete of Host File Cross References

Press Enter to confirm your choices for 4=Delete.
Press F12 to return to change your choices.

Host      -----AS/4 -----
Option  File      File      Library  Host File Description
4      EAMSL5  EAMSALES  QPOSFILES  SMKT ITEM MOVEMENT CURRENT
4      EGRDAY  EAMDAILY  QPOSFILES  SMKT DAILY5 SALES FIGURES
4      DAVES1  DCPHCPTST DCPLIB   TEST FILE FOR HCP EMULATION
4      /MRD 1  PCITEMS  QPOSFILES  ITEM FILE UNTRANSLATED
4      +MRD 1  ITEMMAST  QPOSF     ITEM MASTER GSA HOST TRANSLATE

F12=Cancel

```

Rename Host File Cross Reference

Select option 7 (Rename) from the Work with Host File Cross References menu to access this display.

This display is used to change the host file name in the host file cross-reference list.

```

Rename Host File Cross Reference
To rename host file, type New Name, press Enter.

Host      New
File      Name
+MRD 1    _____

```

HCP Emulation Considerations

These AS/400 files must be in point-of-sale disk image format for the partial Load and Dump commands. HCP emulation does not provide any conversion services as part of its function.

The following HCP keyed record commands are not supported in HCP emulation mode:

- Add Keyed record
- Replace Keyed record
- Read Keyed record
- Delete Keyed record

The AS/400 files must be single-member files with 256-byte physical records. The retail data records are blocked or spanned across the AS/400 256-byte records.

Host File Name Cross-Reference List

HCP emulation uses the host file name cross reference list to determine which AS/400 file the ADCS host system is requesting. ADCS sends a 6-character file name to the HCP emulation program in the AS/400 system. The AS/400 system looks for the 6-character name in its cross-reference file. If it is not there, an error message is returned for the ADCS host system. If the host name is found, the associated AS/400 file and library are then used to respond to ADCS.

HCP Emulation Message Queue

HCP emulation has a message queue (QRTHCPMSG) in library QRAIS. This message queue is provided as an aid to problem analysis of HCP emulation actions. You can display and remove messages from this AS/400 message queue. Messages older than 7 days are automatically removed when HCP emulation is being ended. Each HCP emulation session and the ADCS commands received are logged in this queue.

Retail User Profile Used by HCP Emulation

HCP emulation is run under a special retail user profile (QRAIS). Add Communications Entry (ADDCMNE) command operates under the QRAIS user profile. This profile must be granted object existence authority for the AS/400 files contained in the host file cross-references list. This can be done using the Grant Object Authority

(GRTOBJAUT) command. These are the AS/400 files to be maintained from the ADCS host system.

For more information on using the Grant Object Authority command, see the *CL Reference* book.

See Appendix C, "Utility Trailer Record" for information on the utility trailer record.

Steps Required to Use HCP Emulation

The following steps are required to use HCP emulation:

1. Configure an SNA communications line.

```
CRTLINS DLC LIND(SNUFLINE) RSRNAME(LIN 21) ROLE( SEC) MAXFRAME(265)
```

2. Configure an SNA communications controller.

```
CRTCTLHOST CTLD(SNUFCTL) LINKTYPE( SDLC) APPN( NO)
```

3. Configure an SNA communications device for HCP emulation.

```
CRTDEVSNUF DEVD(SNUFHCP) LOCADR( 1) RMTLOCNAME(SNUFHCP)
HOST( ADCS) RCDLEN(265) BLKLEN(265)
DFTPGM(QRTHCPM) HCPEML(469 )
```

4. Configure an SNA communications device for start user program.

```
CRTDEVSNUF DEVD(SNUFSUP) LOCADR( 3) RMTLOCNAME(SNUFSUP)
HOST( ADCS) RCDLEN(265) BLKLEN(265)
DFTPGM(DUMMY) HCPEML(STRUSRPGM)
```

For more information on configuring lines, controllers, and devices see the *Communications Configuration* book.

5. Create host file cross-reference entries for the AS/400 files to be accessed by the host ADCS program. Use option 3 (Work with host file cross reference) from the Configure Retail Communications Application menu to add new host file names.

6. Add a communications entry to the QRAIS subsystem description.

```
ADDCMNE SBS(D(QRAIS/QRAIS) RMTLOCNAME(SNUFSUP) DFTUSR(QRAIS)
ADDCMNE SBS(D(QRAIS/QRAIS) RMTLOCNAME(SNUFHCP) DFTUSR(QRAIS)
```

7. If the RAIS message file(QRTMSGF) does not reside in the QRAIS library, use the following command to add the Language library to the system library list.

```
CHGSBSD SBS(D(QRAIS/QRAIS) SYSLIBL(CF12924)
```

8. Start the QRAIS subsystem.

```
STRSBS SBS(D(QRAIS/QRAIS)
```

9. Check to see if line, controller, and devices are varied on.

```
WRKCFGSTS CFGTYPE( LIN)
```

10. Start ADCS communications from the host system.

11. Check the QRTHCPMSG queue for verification that host commands have been received.

```
DSPMSG MSGQ(QRAIS/QRTHCPMSG)
```

For more information on using these and other control language commands, see the *CL Reference* book.

HCP emulation is started by a SNA bind request sent from the ADCS host system to the AS/400 system. HCP emulation is ended by a severe error or when an SNA unbind request is received from the host system.

If a SNUF device is configured for both Advanced Data Communications for Stores (ADCS) and retail pass-through, SNUF ignores ADCS and treats the device as a retail pass-through device.

HCP Sense Codes Returned to the Host System

HCP emulation checks each command received from ADCS. HCP sense codes are returned to the ADCS host system for any errors found while checking or performing the requested function. The following list of negative sense codes can be sent by the HCP emulation program:

Sense code Reason and suggested recovery

- | | |
|---------|---|
| X'0101' | Not valid end of data offset received. Change the end of data offset to zero. |
| X'029D' | AS/400 file in use cannot allocate. HCP emulation requires exclusive use of the AS/400 file. Try the ADCS command later. |
| X'02F9' | Insufficient space to allocate file on the AS/400 system. Remove any unnecessary files and libraries from AS/400. |
| X'0707' | Multiple member file not valid for HCP emulation. Specify a single member file for HCP emulation use. |
| X'07C7' | Permanent input/output error on the AS/400 file. Check data validity on the specified AS/400 file. If in error, restore it from your backup media. |
| X'09C2' | AS/400 file/library combination not found. Verify that the file and library combination specified in the host cross-reference exists on the AS/400 system. |
| X'0B9A' | HCP emulation is not authorized to the AS/400 file. Change the authorization on the AS/400 file to include the retail profile. |
| X'0CFA' | File already exists. ADCS requested a create and the file already exists on the AS/400 system. |
| X'0D0D' | Command is not supported by AS/400 HCP emulation. Change the ADCS host command to one supported by HCP emulation. |
| X'1010' | Not valid starting data block (sector) received. The starting data block specified is greater than the number of data blocks in the AS/400 file. Change it to a smaller number and try the command again. |
| X'1111' | Too many data blocks specified. The number of data blocks specified is greater than the number of data blocks in the AS/400 file. Change it to a smaller number and try the command again. |

- | | |
|---------|--|
| X'3F21' | The AS/400 file does not have 256-byte records. The AS/400 file specified in the host cross-reference must be a single member file with 256-byte physical records. Change the host cross-reference to point to the correct file. |
| X'3FFD' | Host file name not found in host cross-reference. Every host file name received is validated with the host cross-reference. Add the host file name desired to the host cross-reference and try the ADCS command again. |

Using Retail File Conversion Displays

The retail file conversion system allows you to convert retail files and to maintain your retail file conversion templates.

Retail Application Interface Suite for AS/400 Main Menu

Use the GO QRAIS/RAISCTL command to display the Retail Application Interface Suite for AS/400 main menu to select retail file conversion.

```

RAISCTL           Retail Application Interface Suite for AS/4
                                     System:  RCHAS4
Select one of the following:

  1. Work with retail communications
  2. Work with retail file conversion
  3. Configure retail communications application
  4. Work with retail logical interface applications

```

Option 2 (Work with retail file conversion)

This option on the main menu displays the Work with Retail File Conversion menu, or the WRKRTLVCVN command can be used to select file conversion.

This is the main retail file conversion menu. It provides access to the retail file conversion and template maintenance displays of the Retail Application Interface Suite.

Work with Retail File Conversion

```

POSCVN           Work with Retail File Conversion
                                     System:  RCHAS4
Select one of the following:

  1. Convert from retail file format
  2. Convert to retail file format
  3. Work with retail conversion templates

```

Option 1 (Convert from retail file format)

This option allows you to convert a retail file to an AS/400 database file.

Option 2 (Convert to retail file format)

This option allows you to convert an AS/400 file to retail format to be loaded to a retail controller.

Option 3 (Work with retail conversion templates)

This option allows you to work with retail conversion templates or create new ones. For example, you can:

- Create conversion templates.
- Change existing conversion templates.
- Delete existing conversion templates.
- Copy one template to another.
- Add, change, or delete records to or from conversion templates.
- Add, change, or delete fields within conversion template records.

Convert from Retail File Format

Select option 1 (Convert from retail file format) from the Work with Retail File Conversion menu to convert a retail format file to an AS/400 file. This conversion changes variable-length fields and records to fixed-length fields and records used by the AS/400 system. Data fields are also converted to either EBCDIC characters or packed numeric. The conversion mapping information is contained in the template file. You can decide to convert only a limited number of records for purposes of testing template changes. The default (*NOMAX) converts all the input records. The conversion can be a lengthy process, therefore you have the option to run it in batch.

```
Convert from Retail File Format
Type choices, press Enter.
Retail template name . . . . . Name, F4 for list
Convert from retail format:
Input file name . . . . . RTLTPLName, RTLTPL
Library . . . . . LIBL Name, CURLIB, LIBL
Input member . . . . . SEARCHName, SEARCH, FIRST
For choice SEARCH:
Selection store . . . . . ALL Number, ALL
Selection date . . . . . PRV Date, ALL
Convert to AS/4 format:
Output file name . . . . . RTLTPLName, RTLTPL
Library . . . . . LIBL Name, CURLIB, LIBL
Output member . . . . . FROMMBName, FROMMBR, FIRST..
Member option . . . . . REPLACEADD, REPLACE, NONE
Number of records . . . . . NOMAX 1-999999, NOMAX
Place on job queue . . . . . Y Y=Yes, N=No
F3=Exit F4=List F5=Refresh F12=Cancel
```

Retail template name

Enter the name of a predefined conversion template to be used for this conversion. If you do not know the name of the template you want to use, press F4 for a selection list.

Input file name

Enter the name of the input file (file in point-of-sale format) to be converted. If you enter the special value *RTLTPL, the system attempts to use the default point-of-sale file name associated with the conversion template. This file must exist before the conversion is requested.

Library

Enter the name of the library in which the input file is located. If the special value *RTLTPL was entered for

the input file, this parameter is ignored. Special values *CURLIB and *LIBL are permitted. They cause the system to resolve the input file specified to your current library or through your current library list. The resolution occurs in your interactive job, thereby causing a fully qualified input file name to be passed to the batch conversion function.

Input member

Enter the name of the database file member containing the data to be converted. The special values *SEARCH or *FIRST can be used. *SEARCH causes the system to use your entries for selection store and selection date to locate your data. *FIRST causes the system to use the first or only member of the input file specified.

Selection store

Enter the store number of the store for which you want to have data converted. If the special value *ALL is specified, then data for all stores found in the file is converted.

Selection date

Enter the date for which you want to convert data. The date should be entered in your system date format. If the special value *ALL is specified, then data for all dates found in the file will be converted.

If you have retail format data members in your input file whose names do not conform to the naming conventions generated by the TOMBR(*STORE) option of the Receive Retail File (RCVRTLF) command, you must either specify a specific member name for retail member or specify *ALL for BOTH selection store and selection date to process your data. Additionally, under this condition if your retail conversion template causes store number and date to be automatically inserted into your converted output, the contents of those fields will be usable, but not able to be predetermined.

Output file name

Enter the name of the AS/400 file into which you want the data to be converted. If the special value *RTLTPL is used, the system attempts to use the default output file identified in the specified conversion template. If the file does not exist before conversion, it will be created by the conversion as a physical file that is not field level.

Library

Enter the library in which the output file is located. If *RTLTPL was specified for output file this entry is ignored. Special values *CURLIB and *LIBL are permitted. They cause the system to resolve the output file specified to your current library or through your current library list. The resolution occurs in your interactive job, thereby causing a fully qualified output file name to be passed to the batch conversion function.

Output member

Enter the name of the database file member that is to contain the converted data. Special values *FROMMBR, *FIRST, and *STORE can also be used. *FROMMBR causes the converted output to be placed into a member that has the same name as the input data member. *FIRST causes the converted data to be placed into the

first or only member of the AS/400 file. *STORE causes the converted data to be placed into a member whose name is based on selection store and selection date. The member name is in the format RIIIIYYJJJ where IIII is the value specified for selection store and YYJJJ is the Julian representation of the value specified for selection date. If the special value *ALL is specified for either selection store or selection date, the respective portion of the member name will be replaced by _ALL.

Member option

Specifying *ADD causes the output of the conversion request to be added at the end of any preexisting data in the output member. Specifying *REPLACE causes any preexisting data in the output member to be cleared prior to the insertion of the results of the conversion.

Specifying *NONE requests the system to place the output in an output member that is not preexisting. If *ADD or *REPLACE is specified, and the output member does not exist, the system will function as if *NONE had been specified.

Number of records

This entry specifies the maximum number of records to be placed in the output file as a result of the conversion. The special value of *NOMAX causes the entire input file to be converted into the output file.

Place on job queue

Enter N to have the conversion run interactively. This is not recommended for all but the smallest files. Enter Y to have your conversion run asynchronously to your work station job and at a lower priority level. This is the recommended response.

Select Retail Conversion Template: Select F4 (List) from the file conversion displays to obtain a list of available retail conversion templates for selection.

```

                Select Retail Conversion Template
Position to . . . _____ Starting characters of template
Type option, press Enter.
  1=Select

Option  Retail      Text
-----  -
-      EALACCTC     GSA Accounting
-      EALDEPTC     GSA Department Totals
-      EALDMCTC     GSA Delayed Maintenance Control
-      EALDMERR     GSA Delayed Maintenance Error file
-      EALDPLST     GSA Department Totals List
-      EALIMLST     GSA Item Movement List
-      EALIMOV     GSA Item Movement
-      EALPERFC     GSA Operator Performance
-      EALTPROD     GSA Terminal Productivity
-      EALTRANS     GSA Transaction Summary Log
-      EAMACCTC     SMA Accounting Totals
-      EAMDMCTC     SMA Delayed Maintenance Control

F12=Cancel
  
```

Enter a 1 next to the retail conversion template you want to use.

Convert to Retail File Format: Select option 2 (Convert to retail file format) from the Work with Retail File Conversion menu to convert an AS/400 file to a retail format file.

```

                Convert to Retail File Format
Type choices, press Enter.

Retail template name . . . _____ Name, F4 for list

Convert from AS/4 format:
Input file name . . . _____ RTLTPL Name, RTLTPL
Library . . . _____ LIBL Name, CURLIB, LIBL
Input member . . . _____ FIRST Name, FIRST

Convert to retail format:
Output file name . . . _____ RTLTPL Name, RTLTPL
Library . . . _____ LIBL Name, CURLIB, LIBL
Output member . . . _____ FROMMB Name, FROMMBR, FIRST
Member option . . . _____ REPLACE ADD, REPLACE, NONE

Output format option . . . _____ SNDRTLFSNDRTLF, SNDRTLRCD
Number of records . . . _____ NOMAX 1-999999, NOMAX

Place on job queue . . . _____ Y Y=Yes, N=No

F3=Exit F4=List F5=Refresh F12=Cancel
  
```

Retail template name

Enter the name of a predefined conversion template to be used for this conversion. If you do not know the name of the template you want to use, press F4 for a selection list.

Input file name

Enter the name of the input file to the conversion. If you enter the special value *RTLTPL, the system attempts to use the default AS/400 file name associated with the conversion template. This file must exist before the conversion request is run.

Library

Enter the name of the library in which the AS/400 file is located. If the special value *RTLTPL was entered for the AS/400 file, this parameter is ignored. Special values *CURLIB and *LIBL are permitted. These values are used by the system to find the AS/400 file specified in your current library or your current library list. This occurs in your interactive job, thereby causing a fully qualified file name to be passed to the conversion.

Input member

Enter the name of the file member within the AS/400 file that contains the data to be converted. The special value *FIRST is permitted. Entering special value *FIRST causes the system to use the first member of the AS/400 file specified.

Output file name

Enter the name of the output file into which you want the data to be converted. If the special value *RTLTPL is used, the system attempts to use the default retail file identified in the specified conversion template. This file must be a physical file with a 256-byte record length. If it does not exist prior to the conversion request, it will be created by the conversion process.

Library

Enter the library in which the output file is located. If *RTLTPPL was specified for output file this entry is ignored. Special values *CURLIB and *LIBL are permitted. These values are used by the system to find the file specified in your current library or your current library list. This occurs in your interactive job, thereby causing a fully qualified file name to be passed to the conversion.

Output member

Enter the name of the file member within the output file in which you want to place the converted data. Special values *FROMMBR and *FIRST are permitted. Entering *FROMMBR causes the converted output to be placed into a member whose name is the same as the AS/400 member used for input. Entering *FIRST causes the data to be placed into the first or only member of retail file.

Member option

The allowed values are the following:

*ADD

This value is for use with preexisting output members and causes the converted data to be added to existing data at the end of the output member.

*REPLACE

This value is also for use with preexisting members and causes all preexisting data to be deleted from the output member prior to the conversion.

*NONE

This value indicates that the output member does not exist and should be created as part of the conversion process.

Note: If *ADD or *REPLACE is specified and the output member does not exist prior to the conversion the system will function as if *NONE had been specified.

Output format option

This parameter specifies whether the output of this command is to be used by the SNDRTLF or SNDRTLRCDC command.

*SNDRTLF

The output of this command is to be used by the SNDRTLF command. Data is placed in the file in PC

format (sector image). Records will be blocked and spanned as appropriate to the file type and controller type specified in the retail conversion template named in parameter RTLTPPL. If the template calls for a keyed file, the conversion will write a direct file that can be sent (using the SNDRTLF command) to the retail controller and then converted to a keyed file by use of the Keyed File Utility.

*SNDRTLRCDC

The output of this command is to be used by the SNDRTLRCDC command. Data is placed in the file in PC format with only a single logical record per physical record. There is no blocking or spanning. This option should only be used if the length of the PC record format is not greater than 256 bytes.

Number of records

This entry specifies the maximum number of records to be placed in the output file as a result of the conversion. The special value of *NOMAX causes the entire input file to be converted into the output file.

Place on job queue

Enter N to have the conversion run interactively. This is not recommended for all but the smallest files. Enter Y to have your conversion run asynchronously to your work station job and at a lower priority level. This is the recommended response.

See Appendix C, "Utility Trailer Record" for information on the utility trailer record.

Conversion Commands

This section contains detailed information on the conversion commands.

Convert To Retail File (CVTTORTLF) Command:

The Convert to Retail File (CVTTORTLF) command causes an AS/400 database file to be converted to point-of-sale file format according to a predefined set of rules in a retail conversion template. This command is the reverse of the Convert From Retail File (CVTFMRRLF) command.

```

CVTTORTLERTLTPL( retail-template-name )
                                RTLTPL
                                LIBL/
                                FROMFILE( file-name )
                                CURLIB/
                                library-name/

                                RTLTPL
                                LIBL/
TOFILE( file-name ) FROMMBR( FIRST
                                member-name )
                                CURLIB/
                                library-name/
                                (P)

                                FROMMBR
                                FIRST
TOMBR( member-name ) MBROPT( NONE
                                REPLACE ) FMTOPT( SNDRTLF
                                ADD )
                                SNDRTLRCD

                                NOMAX
NBRRCD( number-of-records )
  
```

Note:

P All parameters preceding this point can be specified positionally.

Note: Do not attempt conversions where the FROMFILE and TOFILE parameters resolve to the same file.

RTLTPL Parameter

This parameter specifies the name of the retail conversion template to use for the conversion.

FROMFILE Parameter

This parameter specifies the qualified name of the input file in AS/400 format to be converted (library name/file name).

The possible library values are:

***RTLTPL**

Specifies that the qualified input file name is to be retrieved from the retail conversion template specified in the RTLTPL parameter.

file-name

Specifies the name of the input file.

Note: If *RTLTPL is specified for file name the specified library name is ignored. This file must exist prior to issuing the command.

***LIBL**

The library list is used to locate the file name.

***CURLIB**

The current library for the job is used to locate the file name. If no current library entry exists in the library list, QGPL is used.

library-name

Specify the library where the file is located.

TOFILE Parameter

This parameter specifies the qualified name of the output file where retail format data is to be placed (library name/file name).

The possible values are:

***RTLTPL**

Specifies that the qualified output file name is to be retrieved from the retail conversion template specified in the RTLTPL parameter.

file-name

Specifies the name of the output file.

Note: If *RTLTPL is specified for file name, the specified library name is ignored. If this file does not exist prior to issuing this command, it will be created as part of the conversion process.

***LIBL**

The library list is used to locate the file name. If the file is not found the command runs as if *CURLIB had been specified.

***CURLIB**

The current library for the job is used to locate/create the output file. If no current library entry exists in the library list, QGPL is used.

library-name

Specify the library where the file is located.

FROMMBR Parameter

This parameter specifies which member in the input file is to be used for the conversion.

***FIRST**

The first or only member of the input file is used.

member-name

Specify the name of the input file member to be used for the conversion.

TOMBR Parameter

This parameter specifies which member in the output file is to contain the converted data.

***FROMMBR**

Specifies that the output member should have the same name as the reserved FROMMBR parameter.

***FIRST**

The first or only member of the output file is used.

member-name

Specify the name of the output file member to be used for the conversion.

MBROPT Parameter

This parameter specifies whether data in preexisting output members is kept.

***NONE**

Data from the conversion process is not being placed in preexisting output members. The output member will be created as part of the conversion process.

***REPLACE**

Data in preexisting output members is cleared prior to the conversion process.

***ADD**

Data from the conversion process is added to preexisting members after records already in the file.

Note: If *REPLACE or *ADD is specified and the output member does not exist prior to issuing the command, the system will treat it as if *NONE had been specified.

FMTOPT Parameter

This parameter specifies whether the output of this command is to be used by the SNDRTLF or SNDRTLRC command.

***SNDRTLF**

The output of this command is to be used by the SNDRTLF command. Data is placed in the file in PC format (sector image). Records will be blocked and spanned as appropriate to the file type and controller type specified in the retail conversion template named in parameter RTLTP. If the template calls for a keyed file, the conversion will write a direct file that can be sent to the retail controller (using the

SNDRTLF command) and then converted to a keyed file by use of the Keyed File Utility.

***SNDRTLRC**

The output of this command is to be used by the SNDRTLRC command. Data is placed in the file in PC format with one logical record per physical record. There is no blocking or spanning. This option should be used only if the length of the PC record format is not greater than 256 bytes.

NBRRCD Parameter

This parameter specifies the maximum number of records to be placed in the output file.

***NOMAX**

All records selected from the input file will be placed in the output file.

number-of-records

Specifies the maximum number of records to be placed in the output file. If this number is reached, the conversion ends.

The following escape messages can be monitored from this command:

```

POS9501  POS9502  POS9504  POS9505
POS9506  POS9507  POS9508  POS9509
POS9510  POS9511  POS9512  POS9513
POS9514  POS9515  POS9516  POS9517
POS9518  POS9519  POS9520  POS9521
POS9522

```

The following are completion and diagnostic messages that will be placed in the job log by this command:

```

POS9991  POS9992  POS9993  POS9994
POS9995

```

See Appendix C, "Utility Trailer Record" for information about the utility trailer record.

Convert from Retail File (CVTFRMRTLF)

Command: The Convert From Retail File (CVTFRMRTLF) command causes an AS/400 data file in retail format to be converted to an AS/400 database file according to a defined set of rules in a retail conversion template. This command is the reverse of the Convert To Retail File (CVTTORTLF) command.

```

CVTFMRRTLRTLTP( retail-template-name )
                                RTLTPL
                                LIBL/
                                FROMFILE( file-name )
                                CURLIB/
                                library-name/

                                RTLTPL
                                LIBL/
                                TOFILE( file-name )
                                FROMMBR( SEARCH
                                FIRST
                                member-name )
                                CURLIB/
                                library-name/
                                (P)

                                FROMMBR
                                FIRST
                                STORE
                                member-name
                                MBROPT( NONE
                                REPLACE
                                ADD
                                ) DATE( PRV
                                CURRENT
                                ALL
                                date
                                )

                                ALL
                                STORE( store-identifier )NBRRCD( NOMAX
                                number-of-records
                                )
  
```

Note:

^P All parameters preceding this point can be specified positionally.

Note: Do not attempt conversions where the FROMFILE and TOFILE parameters resolve to the same file.

RTLTP Parameter

This parameter specifies the name of the retail conversion template to use for the conversion.

FROMFILE Parameter

This parameter specifies the qualified name of the input file in retail format to be converted (library name/file name).

The possible values are:

***RTLTP**

Specifies that the qualified input file name is to be retrieved from the conversion template specified in the RTLTP parameter.

file-name

Specifies the name of the input file.

Note: If *RTLTP is specified for file name, the specified library name is ignored. This file must exist before issuing the command.

***LIBL**

The library list is used to locate the file name.

***CURLIB**

The current library for the job is used to locate the file name. If no current library entry exists in the library list, QGPL is used.

library-name

Specifies the library where the file is located.

TOFILE Parameter

This parameter specifies the qualified name of the output file where AS/400 format data is to be placed (library name/file name). This file must be a physical file with record length of 256 bytes.

The possible values are:

***RTLTP**

Specifies that the qualified output file name is to be retrieved from the conversion template specified in the RTLTP parameter.

file-name

Specifies the name of the output file.

Note: If *RTLTP is specified for file name, the specified library name is ignored. If this file does not exist prior to issuing the command it will be created as part of the conversion process.

***LIBL**

The library list is used to locate the file name. If the file is not found the command runs as if *CURLIB had been specified.

***CURLIB**

The current library for the job is used to locate/create the file name. If no current library entry exists in the library list, QGPL is used.

library-name

Specifies the library where the file is located.

FROMMBR Parameter

This parameter specifies the members of the input file from which data is to be retrieved for conversion.

***SEARCH**

Values specified in parameters DATE and STORE are to be used.

***FIRST**

The first or only member of the input file is to be used.

member-name

The name of the single input member to be used.

TOMBR Parameter

This parameter specifies the members in the output file to contain the converted data.

***FROMMBR**

Converted data is placed into the members whose names are the same as the members from which the data came.

***FIRST**

The first or only member in the output file is used for all converted data.

***STORE**

The data is placed into a member whose name is based on the DATE and STORE parameters. The member is of the form RIIIIYYJJJ where IIII is the store id specified in the store parameter and YYJJJ is the julian representation of the value specified in the DATE parameter. If either DATE or STORE is specified as *ALL, its portion of the member name is replaced by ____ALL.

member-name

The name of the member into which the converted data is placed.

MBROPT Parameter

This parameter specifies whether data in preexisting output members should be kept.

***NONE**

Data is not being placed in preexisting output members. The conversion creates the output members. If the conversion is specified to have multiple input members, FROMMBR(*SEARCH), and to have a single output member, TOMBR(*FROMMBR), **not** specified, conversion of input members subsequent to the first will be processed as if MBROPT(*ADD) had been specified.

***REPLACE**

Data in preexisting output members should be cleared before the converted data is placed in the member. If the conversion is specified to have multiple input members, FROMMBR(*SEARCH), and to have a single output member, TOMBR(*FROMMBR), **not** specified, the data in the output member is cleared only for the first input member. Conversion of subsequent input members will be processed as if MBROPT(*ADD) had been specified.

***ADD**

Data in preexisting output members is kept and converted data is placed into those members after the existing data.

Note: If *REPLACE or *ADD is specified and the output member does not exist prior to issuing the command, the system will function as if *NONE had been specified.

DATE Parameter

This parameter specifies the date for which you want to convert retail data.

***PRV**

The date used is the calendar day prior to the current job date.

***CURRENT**

The date used is the current job date.

***ALL**

All dates found in the file are to be processed.

date

Specify the date to process. The date should be entered in the format as specified by system value QDATFMT.

STORE Parameter

This parameter specifies the store for which retail data is to be converted.

***ALL**

Data for all stores is to be processed.

store-identifier

The store-id for which data is to be processed.

If you have retail format data members in your input file (FROMFILE) whose names do not conform to the naming convention generated by the *STORE option of the RCVRTLF command you must either specify FROMMBR (member-name) or DATE(*ALL) and STORE (*ALL) to process your data. In this case all members in your input file will be processed. If your retail conversion template causes store-ID and date to be automatically inserted into your converted output, the contents of those fields cannot be determined if your member naming convention does not conform to that of TOMBR(*STORE) of the RCVRTLF command.

NBRRCDs Parameter

This parameter specifies the maximum number of records to be placed in the output file.

***NOMAX**

All records selected from the input file are placed in the output file.

number-of-records

Specifies the maximum number of records to be placed in the output file. If this number is reached, the conversion ends.

FROMMBR(*SEARCH) Option

The FROMMBR(*SEARCH) instructs the conversion code on how to locate the input members of FROMFILE for conversion. *SEARCH is only appropriate if the members of FROMFILE are named according to the convention established by RCVRTL F TOMBR(*STORE). If *SEARCH is specified here, the values provided by parameters DATE and STORE will be used to match to the member named according to the RXXXXYYJJJ format, where xxxx is the store number and YYJJJ is the Julian date. Date processing (*PRV and *CURRENT) is determined by the job date (obtained through use of the RTVJOBA command). This can cause multiple input members to any given conversion.

TOMBR(*STORE) Option

The TOMBR(*STORE) defines member names to be used for the output of the conversion. TOMBR(*STORE) causes the output member to be named RXXXXYYJJJ.

When using naming conventions that are different than the naming convention established by the RCVRTL F command TOMBR(*STORE) option, do not use the FROMMBR(*SEARCH) option of the CVTFMRMRTL F command. Specify either *FIRST or the member name for FROMMBR to access this data. If you want to have multiple input member capability for a conversion, specify FROMMBR(*SEARCH), but also specify DATE(*ALL) and STORE(*ALL). You must then ensure that only data members that you want converted exist in FROMFILE. If you use TOMBR(*STORE), your data is placed into output member R_ALL_ALL.

Both the DATE and STORE parameters are used for name checking of the input members to the conversion. There is no relationship between file conversion and the store information file. DATE(*ALL) and STORE(*ALL) indicate to the conversion driver that no name checking of input members will occur if FROMMBR(*SEARCH) is specified.

The following escape messages can be monitored from this command:

POS9501	POS9502	POS9503	POS9504
POS9505	POS9506	POS9507	POS9508
POS9509	POS9510	POS9511	POS9512
POS9513	POS9514	POS9515	POS9516
POS9517	POS9518	POS9519	POS9520
POS9521	POS9522		

The following completion and diagnostic messages are placed in the job log by this command:

POS9991 POS9992 POS9993 POS9994
 POS9995

Conversion Command Data Conversion Error Log:

The file translation system will produce an error log of data records that it cannot translate. Contained within each error-log record is identifying information about the input and output to the translation as well as an error code and the input data. The format of this log can be determined by issuing command DSPFFD QRAIS/QAZCFXERR. The actual error log will be in library QUSRSYS, and the specific member containing the error data for your conversion will be identified in the POS9995 message help, when issued. If message POS9995 is **not** issued as a result of your conversion, no records were placed in the error log. The format of the data in the error log is detailed in the following table.

Table 4-3. Format of Data in Error Log

From	To	Contents
1	10	Conversion job name
11	20	Conversion job user ID
21	26	Conversion job number
27	36	Input file library
37	46	Input file name
47	56	Input file member name
57	66	Output file library
67	76	Output file name
77	86	Output file member name
87	96	Template name
97	100	Conversion error code 4 position character will contain the error code for data errors. Refer to the <i>Retail Communications Programming</i> for more information. For unknown record ID, this field will contain zeros.
101	101	Unknown record ID flag, Y=unknown record ID
102	106	Record ID of record in error
107	123	RESERVED
124	128	Field sequence number from translation template for field containing incorrect data (5 positions, 0 decimals zoned)
129	1637	Input record received for translation.

Note: Data in positions 102 through 1637 will be actual data supplied to the conversion for unknown record ID errors. For incorrect data errors, the record will be in a fixed format, as described in the retail translation template used with all delimiters removed.

Chapter 5. RAIS Retail Polling Application

This chapter gives you the information needed to work with the RAIS Retail Polling Application option (option 2).

The Retail Polling Application uses the RAIS communications API to process retail application files as the files get updated by the retail controller application. A Polling entry must first be configured for each of the file and store combinations that a user wishes to process. Each entry must have a unique Store Number, Template, and File Table.

The Polling application will retrieve the specified file data from the locations referenced by the Store Number field. Store Number is defined using the Work with Retail Store Information screen. See “Configuring for Retail Communications” on page 2-1 for information on how to configure a store. The file that is accessed is defined in the Retail File Table. The File Table is a sequenced list of Retail Application Files. The list is ordered so that the first is the Active Retail Application file, and all others are archived Retail Application files. The polling application will continuously retrieve data from the Active file until the file does not exist, or the date in the file changes. In these cases, the polling application will check if the Active file has been archived by matching the last processed data with offset data in the archived files. If a match occurs, and the archived file has additional data that has not been processed, the polling application will continue processing the archived file. After processing the rest of the data, the polling application will process the active file. See the “Retail File Table Configuration” on page 5-5 for information on how to configure the Retail File Table.

During processing the Template support is used to convert the Retail Data retrieved into an AS/400 usable form. The data can be converted to an AS/400 file just as the Convert From Retail File command (CVTFMRRLF) does, or it can also be converted and placed in a predefined relational database collection. See “Work with Retail Conversion Templates” on page 2-7 for information on how to configure the Templates.

The Work with Retail Polling Entries screen is used to configure the polling entries.

Work with Retail Polling Entries

```
Work with Retail Polling Entries
Position to . . . . . Starting characters of Store Number
Type options, press Enter.
  2=Change  4=Delete  5=Display  9=Start  1 =End
Option  Store Number  Template  File Table  Offset  Status
  2      111111  EALTRANS  RTBL111111  ENDED
Bottom
F3=Exit    F5=Refresh  F6=Add Retail Entry
F12=Cancel F14=Work with Retail File Table  F21=Print list
```

See “Retail Polling Entries Field Descriptions” on page 5-2 for an explanation of the data on this display.

Change Retail Polling Entry: Select option 2 (Change) from the Work with Retail Polling Entries display to change the polling entry.

```
Change Retail Polling Entry
Type changes, press Enter.
Store number . . . . . 111111
Template to use . . . . . EALTRANS
Retail file table . . . . . RTBL111111
Autostart entry . . . . . N Y or N
F3=Exit  F12=Cancel
```

From the change screen you are allowed to change the Autostart Entry only. See “Retail Polling Entries Field Descriptions” on page 5-2 for an explanation of the data on this display.

Display Retail Polling Entry: Select option 5 (Display) from the Work with Retail Polling Entries display to see polling entry configured values.

```

Display Retail Polling Entry
Store number . . . . . : 111111
Template to use . . . . . : EALTRANS
Retail file table . . . . . : RTBL111111
Autostart entry . . . . . : N

Press Enter to continue.
F3=Exit  F12=Cancel

```

```

Job: B,I Pgm: B,I REXX: B,I Exec
Threadsafe: No

ADDRTLPLLE
STORENBR( retail-store-number )
TEMPLATE( conversion-template-nameP )
FILETABLE( retail-file-table )

AUTOSTART( N Y PROCONCE( N Y ) )

```

Note:
^P All parameters preceding this point can be specified positionally.

See "Retail Polling Entries Field Descriptions" for an explanation of the data on this display.

Add Retail Polling Entry: Press F6 (Add polling entry) from the Work with Retail Polling Entries display to start the Add Retail Polling Entry Command.

The remove retail polling entry command is used to remove polling entries from the polling entry configuration table.

```

Add Retail Polling Entry (ADDRTLPLLE)
Type changes, press Enter.

Store Number . . . . . 1-999999
Template to use . . . . . Character value
Retail file table . . . . . Character value
Autostart entry . . . . . N Y, N
Process Entry Once . . . . . N Y, N

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

```

Job: B,I Pgm: B,I REXX: B,I Exec
Threadsafe: No

RMVRTLPLLE
STORENBR( retail-store-number )
TEMPLATE( conversion-template-nameP )
FILETABLE( retail-file-table )

```

Note:
^P All parameters preceding this point can be specified positionally.

Retail Polling Entries Field Descriptions

See "Retail Polling Entries Field Descriptions" for an explanation of the data on this display.

Confirm Delete of Retail Polling Entries: Select option 4 (Delete) from the Work with Retail Polling Entries display to confirm deletion of retail polling entries.

Store Number

This is the store number of the Retail Application Controller that the polling application will use to access the file specified in the Template. The store number location must be configured on the Work with Retail Communications menu for the polling application to successfully connect to the store controller. See "Configuring for Retail Communications" on page 2-1 for information on how to configure your store controller.

Template

This is the name of the retail template which will be used to convert the retrieved retail data. The template header description should also have the default Remote Path configured. The remote path is the path and file name on the retail controller that will be read during the polling application processing of this entry. See "Work with Retail Conversion Templates" on page 2-7 for a information on how to configure the template description.

File table

This is the name of the retail file table which is a list of path specified retail application file names. The polling application uses the file names as the source of the retail data conversion. See "Retail File Table Field

```

idssname uimname
Confirm Delete of Retail Polling Entry

Press Enter to confirm your choices for 4=Delete.
Press F12 to return to change your choices.

Store      File
Option  Number  Template  Table  Status
4  111111  EALTRANS  RTBL111111  ENDED

F12=Cancel

```

Retail Polling Entry Configuration commands: The add retail polling entry command is used to add polling entries to the polling entry configuration table.

Descriptions” on page 5-7 for a information on how to configure the retail file table and its entries.

Autostart entry

This is a configurable parameter which is used to tell the polling application if this entry should automatically be started when the polling application is started.

N=NO means *not* to start this entry automatically.

Y=YES means to start processing this entry automatically.

Process Entry Once

This is a configurable parameter which is used to tell the polling application if this entry should only be processed until end of file is reached.

N=NO means *not* to end this entry automatically.

Y=YES means to end processing of this entry at end of file.

Status (WRKRTLPLLE only)

This is status field on the Work with Retail Polling Entries screen. It shows the current status of the entry. The possible values are:

Ended

The polling entry is in an ended state. The polling application is not currently processing the entry.

Ending

The ENDRTLPLLE command was processed for this entry. The polling application is currently processing the entry and will end the entry when processing completes.

Started

The STRRTLPLLE command was issued for this entry. The polling application has not started processing this entry.

Running Primary

The polling application is currently processing the entry using the primary controller connection.

Running Alternate

The polling application is currently processing the entry using the alternate controller connection.

Running - *error_data*

An error was encountered while processing this entry. The polling application did not end this entry. One of the following has occurred:

A communications error was encountered while processing this entry.

The ***error_data*** field will be set to one of the communications error code values. See “AS/400 UNIX Error Codes” on page E-1 for additional information regarding the error codes.

The polling application will continue to attempt communications with this store controller. If this error continues, check the status of the store controller connections to determine the error.

When the controller is reconnected, the Communications Error will be reset.

A data conversion error was encountered while processing this entry.

The ***error_data*** field will be set to ECONVERT(3490). A data conversion error was encountered while processing this entry. The polling application will continue processing. Check the QAZCFDBERR file in QUSRSYS to help determine the error. Clear the member associated with this error. End the Retail polling entry and restart to remove the data error indication.

Error - *error_data*

An error was encountered while processing this entry that caused the polling application to end this entry. One of the following has occurred:

A communications error was encountered while processing this entry.

The ***error_data*** field will be set to one of the communications error code values. See “AS/400 UNIX Error Codes” on page E-1 for additional information regarding the error codes.

The polling application will continue to attempt communications with this store controller. If this error continues, check the status of the store controller connections to determine the error. When the controller is reconnected, the Communications Error will be reset.

A data conversion error was encountered while processing this entry.

The ***error_data*** field will be set to ECONVERT(3490). A data conversion error was encountered while processing this entry. The Last conversion caused the QAZCFDBERR file to become full. This caused the polling application to stop processing. Check the QAZCFDBERR file in QUSRSYS to help determine the error. Clear the member that has become full. End the Retail polling entry and restart to remove the data error indication.

The template definition is not correctly defined for the polling application. The ***error_data*** field will be set to ETERM(3464).

The polling application could not allocate a system resource. The ***error_data*** field will be set to ETERM(3464).

The polling application could not allocate a configuration file needed to run this entry. The ***error_data*** field will be set to ETERM(3464).

Use the retail polling entry(ENDRTLPLLE) command to end the polling entry. Correct the error that occurred and restart the retail polling entry(STRRTLPLLE).

Polling Application Commands

The polling application is controlled using the following commands.

- Start Retail Server (STRRTLSVR)
- End Retail Server (ENDRTLSVR)
- Start Retail Polling Entry (STRRTLPLLE)
- End Retail Polling Entry (ENDRTLPLLE)

Start Retail Server: The start retail server command is used to start the Retail application server subsystem (QZCFSERVER) and optionally start any retail polling entries that are configured as Autostart Yes.

```

Job: B,I Pgm: B,I REXX: B,I Exec
Threadsaf: No

STRRTLSVR (P)

```

Note:
^P All parameters preceding this point can be specified positionally.

End Retail Server: The end retail application server command is used to end the active Retail polling entries and applicatiuon serving subsystem.

```

Job: B,I Pgm: B,I REXX: B,I Exec
Threadsaf: No

ENDRTLSVR (P)

```

Note:
^P All parameters preceding this point can be specified positionally.

Start Retail Polling Entry: The start retail polling entry command is used to start polling on the defined entry.

```

Job: B,I Pgm: B,I REXX: B,I Exec
Threadsaf: No

STRRTLPLLE
STORENBR( ALL
          retail-store-number
          )
RTLTPL( ALL (P)
         conversion-template-name
         )
RTLFILETBL( ALL
             retail-file-table
             )
PROCONCE( N
           Y
           )
OFFSET( LAST
        retail-file-offset
        )

```

Note:
^P All parameters preceding this point can be specified positionally.

STORENBR Parameter

This is the store number part of the retail polling entry that is to be started. Possible values are:

store-number

A specific store number will be started.

*ALL

All store entries with the template name defined in the RTLTPL parameter and file table name defined in the RTLFILETBL parameter will be started.

RTLTPL Parameter

This is the template name part of the retail polling entry that is to be started. Possible values are:

template-name

A specific store entry with this template name will be started.

*ALL

All store entries with the store number defined in the STORENBR parameter and file table name defined in the RTLFILETBL parameter will be started.

RTLFILETBL Parameter

This is the file table name part of the retail polling entry that is to be started. Possible values are:

*ALL

All store entries with the store number defined in the STORENBR parameter and the template name defined in the RTLTPL parameter will be started.

file-table-name

A specific store entry with this file table name will be started.

PROCONCE Parameter

This is the parameter which is used to tell the polling application if this entry should only be processed until end of file is reached.

N

Polling application will *not* end this entry automatically.

Y

Polling application will end this entry automatically when end of file is reached.

OFFSET Parameter

This is the retail file offset. The polling application uses this offset to start reading the retail file.

***LAST**

The last retail file offset will be used by the polling application. The polling application keeps an offset into the retail file where it last processed data. This offset is used.

retail-file-offset

A specific offset into the retail file is used by the polling application.

Note: Please use this with caution. The offset may need to point to a record level offset, or unexpected results may occur.

End Retail Polling Entry: The end retail polling entry command is used to indicate the polling application should end processing on this entry. The polling application will end the polling when processing is complete.

```

Job: B,I Pgm: B,I REXX: B,I Exec
Threadsaf: No

ENDRTLPE
STORENBR(      ALL          )
           retail-store-number
RTLTPPL(      ALL          (P) )
           conversion-template-name

           ALL
RTLFILETBL(   file-table name )

Note:
P All parameters preceding this point can be specified
positionally.

```

STORENBR Parameter

This is the store number part of the retail polling entry that is to be ended. Possible values are:

store-number

A specific store number will be ended.

***ALL**

All store entries with the template name defined in the RTLTPPL parameter will be ended.

RTLTPPL Parameter

This is the template name part of the retail polling entry that is to be ended. Possible values are:

template-name

A specific store entry with this template name will be ended.

***ALL**

All store entries with the store number defined in the STORENBR parameter will be ended.

RTLFILETBL Parameter

This is the file table name part of the retail polling entry that is to be started. Possible values are:

***ALL**

All store entries with the store number defined in the STORENBR parameter and the template name defined in the RTLTPPL parameter will be started.

file-table-name

A specific store entry with this file table name will be started.

Retail Polling Application Notes

The polling application is designed for use with sequential string files.

For the non-relational database support, the polling application will create the database member with a name Rxxxxyyddd, where;

xxxx

4 character store number from WRKRTLPLLE screen.

yy

2 character year of time conversion is called within the polling application.

ddd

Julian day of the year when conversion is called within the polling application.

Note: This is different from the RCVRTLFL command where the yyddd date is taken from the job.

Retail File Table Configuration

In order to run the polling application, the File Table must exist and have at least one Retail File name defined. The retail File table is a list of Retail Application files that reside on the Retail application controller. This can be built as a sequential list for those Retail controller applications that archive transaction data when a store closes. This list allows the polling application to search the archived files to continue to read the current data that has now been archived.

Work with Retail File table: Press F14 (Work with Retail File Table) from the Work with Retail Polling Entries display to work with this list of retail file tables.

```

Work with Retail File Table
Position to . . . . . Starting characters of File Table
Type options, press Enter.
 2=Change  4=Delete  5=Display
 8=Work with Retail File Table Entries

File
Option  Table      Description
 5      RTBL111111  The retail file table for store 111111

F3=Exit  F5=Refresh  F6=Create Retail File Table
F12=Cancel

```

```

Display Retail File Table
Type changes, press Enter.
Retail file table name . . . . . : RTBL111111
Retail file table description . . . . . The retail file table for 111111

F3=Exit  F12=Cancel

```

Work with Retail File Table Entries: Select Option 8 (Work with Retail File Table Entries) from the Work with Retail File Table display to work with this list of a retail file table entries.

Create Retail File Table: Press F6 (Create Retail File Table) from the Work with Retail File Table display to create a retail file table.

```

Create Retail File Table
Type changes, press Enter.
Retail file table name . . . . .
Retail file table description . . .

F3=Exit  F12=Cancel

```

```

Work with Retail File Table Entries
Type options, press Enter.
 2=Change  4=Delete  5=Display

Sequence
Option  Number  Retail Application File
 5      1      C:\ADX_IDT1\EALTRANS.DAT

F3=Exit  F5=Refresh  F6=Add Retail File Table Entry
F12=Cancel

```

Add Retail File Table Entry: Press F6 (Add Retail File Table Entry) from the Work with Retail File Table Entries display to create a retail file table.

Change Retail File Table: Select option 2 (Change) from the Work with Retail File Table display to change the a retail file table description.

```

Change Retail File Table
Type changes, press Enter.
Retail file table name . . . . . : RTBL111111
Retail file table description . . . . . The retail file table for 111111

F3=Exit  F12=Cancel

```

```

Add Retail File Table Entry
Retail file table. . . . . : RTBL111111
Type changes, press Enter.
Sequence number . . . . .
Retail application file . . . . .

F3=Exit  F12=Cancel

```

Change Retail File Table Entry: Select option 2 (Change) from the Work with Retail File Table Entries display to change the a retail file table description.

Display Retail File Table: Select option 5(Display) from the Work with Retail File Table display to display a retail file table description.


```

Change Retail File Table Entry

Retail File Table. . . : RTBL111111
Type changes, press Enter.

Sequence number . . . . . : 1
Retail application file . . . . . : C:\ADX_IDT1\EALTRANS.DAT

F3=Exit  F12=Cancel

```

Display Retail File Table Entry: Select option 5(Display) from the Work with Retail File Table Entries display to display a retail file entry description.

```

Display Retail File Table Entry

Retail file table. . . : RTBL111111
Sequence number . . . . . : 1
Retail application file . . . . . : C:\ADX_IDT1\EALTRANS.DAT

F3=Exit  F12=Cancel

```

Retail File Table Field Descriptions

Retail Application File

This is the path/file name to the retail data. This is used by the retail polling application. The polling application will read this file on the retail controller or in the AS/400 IFS. For Path/File names that are not first in the sequence, the Polling application will consider these archived files. If the polling application attempts to read the file specified in the list, and determines that the file no longer exists, or is not the same file(Date has changed), the archive list is checked to determine if any data from the now archived file needs to be processed. If the polling application does determine the archived data need to be processed, processing will continue until the polling application reads end of file. The polling application will then continue processing the lowest sequenced file.

Retail File Table

This is the name of the Retail File Table. This table holds the list of Retail Application Path/File names.

Sequence Number

This is the sequence of the retail application file name. The file names are ordered with the lowest being the active retail application file. All other files are archived retail application files.

Retail Application File Considerations

The retail application file honors the 6 character HCP file naming convention. The processing will attempt to convert the 6 character file and will replace the name if possible. If the file name cannot be converted, the file is assumed to be a stream file. It is recommended that the file name used not be 6 characters unless it is a HCP logical file.

Polling Application Exit Program Considerations

The Polling Application has the ability to call a user Exit program with a pointer to the current converted data. The Exit program name is configured using the Record Template support. See "Retail Conversion Template Record Field Descriptions" on page 2-13 for information on configuring the retail record exit program. The Polling application will pass the following parameters to the Exit program.

Table 5-1. Format of User Exit Parameters

From	To	Contents
1	4	(BINARY 4) Number of bytes in buffer
5	32766	Buffer of converted data

Polling Application Job Description

The retail polling application uses the QRAIS/QZCF POLL job description. The user can update the job description to add customer library lists to the job. This would be used during exit program processing. The user may also change the run priority of the QRAIS/QZCF POLL class description if they want to give the polling application a higher run priority, or time slice.

Polling Application Performance Considerations

If the retail polling application is being used to insert data into journalled relational database files, this will increase the disk activity on your system and may have a noticeable impact on system performance. One way to improve this performance is to isolate journal receivers in a user ASP. This reduces contention when accessing the disks. See the *OS/400 Backup and Recovery* manual for additional information on User ASPs, Journals and Journaling performance.

Once the User ASP is configured do the following to isolate your journal receiver:

- Create a library in the user ASP if one does not already exist.
- CRTLIB LIB(RCVRLIB) ASP(2)
- Create the new journal receiver in this library.
- CRTJRNRCV JRNRCV(RCVRLIB/DBCOLLECT) ASP(2)
THRESHOLD(1919999)
- Associate the Journal with the new receiver.

CHGJRN JRN(DBCOLLECT/QSQJRN) JRNRCV(RCVRLIB/QZCFGXXXX) = Store number specified for the controller on the WRKRTLPLE entry

Note: The relational database collections installed with options 1 of this product are created with the following journals:

Option 1 General Sales Application
QUSRPOSGS/QSQJRN

Note: Since the option 1 journal is used only when RMVRTLTNS command is used, there are no performance impacts.

Polling Application Authority Considerations

The polling application runs in the QZCFSEVER subsystem under the QRAIS user profile. The following authorities must be granted to this user profile or an EACCES(3401) error will be shown in the status field.

For processing polling entries with a defined exit program, the profile must be granted *USE authority to the exit program.

For processing polling entries with no relational database defined, the user profile must be granted *ADD to the libraries specified on the Retail file and AS/400 file fields in the template. If the files exist, the user profile must be granted *OBJMGT or *OBJALTER authority to these files.

For process polling entries with the relational database defined, the user profile must be granted *W access to the database table as specified by the Data file parameter in all the template record entries. The user profile must be granted *X access to the Database Collection as specified by the Data library parameter in the template header entry.

For processing polling entries that have an IFS location specified, the user profile must be granted *X access to each directory in the path name preceding the file to be read. The user profile must be granted *R access to the file.

The QRAIS user profile is made the owner of the QUSRPOSGS relational database collection that is installed with option 10. So no additional authorization is needed for QRAIS to insert transaction data into this default relational database collection unless user defined tables are added.

Polling Application Data Conversion Error Log:

For the polling application conversion the QUSRSYS/QAZCFDBERR file will be updated with the appropriate error logging information. When polling entries that have a template that has a defined data library, the error log member created will be named with the defined DB2 Collection. For templates without a DB2 Collection defined, the member created will be named:

RXXXXYYJJJ: where
R = Retail

YYJJJ = Julian date that the file is converted

The raw 4690 data will be sent to the error log.

For the DB2 Collection defined templates there is 2 potential errors. The first is the raw retail data is not defined in the template, and the second possibility is the converted data could not be inserted into the DB2 Tables. For this will be 2 entries for the problem data. The first will be the RAW retail data format, and the second will be the converted data format. The error data entries will be structured as the following:

Table 5-2. Format of API Data in Error Log

From	To	Contents
1	6	Store ID
7	16	DB2 Table name(Physical file)
17	26	Conversion template record used
27	256	Input record received for translation, or the converted data.

Note: No message will be sent. The polling application will reflect the error in the status information on the Work with Polling Entries screen. See "Retail Polling Entries Field Descriptions" on page 5-2 for information pertaining to the polling entry status.

Polling Application Status Via Program Call

There is a program which can be invoked to return whether the retail polling application is active or not. The program is QCFPOLSTS in the QRAIS library. This program will issue a POS9050 informational message if the polling job is active and a POS9040 informational message if the polling job has ended. In addition to the informational messages, the program returns a 0 if the polling job has ended and greater than 0 if the polling job is running.

qcfpolsts - Get Retail Polling Status

Syntax

```
int qcfpolsts();
call qcfpolsts
```

Parameters:

none

Return Value(Binary 4):

- 0 The polling job has ended. (POS9040 message)
- 1 The polling job is running. (POS9050 message)
- 1 The polling job is in an error condition.

Chapter 6. Retail Application Logical Interface

This chapter gives you the information needed to work with the RAIS Retail Application Logical Interface(RALI) options (option 11).

The Retail Application Logical Interface(RALI) is a logical view interface to the RAIS relational database options. The information in these logical views represent usable data that can be easily extracted from any of the predefined RAIS relational database collections. These logical views should shield the user from the retail application used to propagate the database.

RALI Relational database considerations

Most logical views of the Database collection remove all post voided and VPLI voided amounts from the totals. Since this is the case, the VPLI and post void transaction must accurately reflect the voided data. This includes correct operator, transactions number, terminal number and date in some cases. If the voided data is not correctly inputted, the views will likely include voided data.

GSA

1. The **General Sales Application** VPLI function does not set the original transaction date and time. The VPLI header date and time will be used. A VPLI issued on a different day than the original transaction will not be reflected in these views. Also any discounts and allowances must correctly be voided during a VPLI void transaction.
2. The GSA offline function may include the transaction strings for the last offline transaction. The view will not include any data from these strings when if the data is reentered, with the exception of the TRANAMOUNT view.
3. The views of the GSA reentry function will not update the transaction number of the original transaction.

Most of the logical views reflect data from several of the transaction strings not updated with the standalone(offline) transaction summary data. Standalone (offline) transactions should be reentered to correctly reflect these transactions.

Populating Optional Relational database.

The QUSRPOSGS relational database collection is created when RAIS option 10 is installed. This collection can be populated using the polling application and defining a polling entry that uses the QZCFGSDDB template definitions. The Retail file table should be defined using the GSA TLOG file name as specified in the 4680-90 programming guides. See Appendix F, "**General Sales Application** Cross Reference" on page F-1 for information on how the QUSRPOSGS tables are populated.

The QUSRPOSSA relational database collection is created when RAIS option 11 is installed. This collection can be populated using the polling application and defining a polling entry that uses the QZCFSADB template definitions. The Retail file table should be defined using the SA TLOG file name as specified in the 4680-90 programming guides. See Appendix G, "**Supermarket Application** Cross Reference" on page G-1 for information on how the QUSRPOSSA tables are populated.

Once polling has populated the collection, the logical interface may be used to extract the complex data.

RALI Views

Transaction Views

"TRANAMOUNT - Transaction Amount Table" on page 6-3

"HEADER - Header Key Table" on page 6-4

"MDSETOTALS - Transaction Merchndise Totals Amount" on page 6-5

"TAXAMT - Transaction Tax " on page 6-6

"TAXCODE - Transaction Tax Tax Code " on page 6-7

"DISCOUNT - Transaction Discounts" on page 6-8

"TENDER - Transaction Tender Amounts" on page 6-9

"TENDERCHG - Amount of Tender Change" on page 6-10

"TRANSVOID - Sales on Void Transaction" on page 6-11

"POSTVOID - Sales on Post Void Transactions" on page 6-12

"DISCREPANCY - Transaction Discrepancy Table" on page 6-13

Merchandise Views

"MDSERSALES - Merchandise Sales Table" on page 6-15

"LAYMDSE - Layaway Merchandise Sales Table" on page 6-16

"ITEMDISCOUNT - Item Discounts Given" on page 6-17

"ITEMDEPOSIT - Item Deposits" on page 6-18

"ITEMVOID - Sales on Voided Items" on page 6-19

"VPLVOID - Sales of Line Item Voids" on page 6-20

Other Views

"ISSUETNDR - Gift Certificate and Store Credit Issued" on page 6-21

"ACCTNEW - Total Amount of New/Adjusted Accounts" on page 6-22

"ACCTPAY - Total Amount of Account Payments" on page 6-23

"ACCTDEP - Total Amount of Account Deposits" on page 6-24

"ACCTFEE - Total Amount of Fees on Accounts" on page 6-25

"LAYCAN - Total Amount of Cancels on Layaways" on page 6-26

"LAYRSTK - Total Amount of Restocks on Layaways" on page 6-27

"PICKUPS - Transaction Pickup Amounts" on page 6-28

"PAYOUTS - Transaction Payout Amounts" on page 6-29

"PAYINS - Transaction Payin Amounts" on page 6-30

"TNDRCASH - Tender Cashed Amounts" on page 6-31

"SPRSNSALES - Total Amount of Salesperson Sales" on page 6-32

"SPECIALSALES - Special Sales Amounts" on page 6-33

TRANAMOUNT - Transaction Amount Table

This view gives a view of the transaction amounts information for each transaction. All positive and negative amounts (such as refunds, allowances, or discounts) are entered or calculated.

<i>Table 6-1. TRANHEADER - Transaction Header Table</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database.
TranNum	integer	Transaction number as it appears from the terminal.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal.
FuncType	integer	The transaction function type. See "Valid Types of Transaction Functions" on page H-3 for descriptions of the function types.
GrossPos	decimal(15,0)	The total amount of positive entries for this transaction.
GrossNeg	decimal(15,0)	The total amount of negative entries for this transaction.
NetCash	decimal(15,0)	Net cash totals for sales transactions. This may be a negative value.
NetNonCash	decimal(15,0)	Net Non-cash totals for sales transactions. This may be a negative value.

HEADER - Header Key Table

This view gives a view of the Key header information for each transaction.

<i>Table 6-2. HEADER - Header Key Table</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database.
TranNum	integer	Transaction number as it appears from the terminal.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal.

MDSETOTALS - Transaction Merchndise Totals Amount

This view gives a summary of the taxable and nontaxable merchandise amounts. The information in this view contains the following:

- A summary of the merchandise sales(minus merchandise specific discounts) that were taxable and non-taxable for each transaction.

- A summary of the transaction discounts for sales that were taxable and non-taxable for each transaction.

- A summary of transaction fees that were taxable and non-taxable for each transaction.

The summary excludes post voids, voids, cancels, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

It will also remove merchandise discounts and allowances and all transaction discounts and fees. The RETURNED flag specifies the totals that were for returns.

Note: VPLI voided items will show up in the view item listing. The amount and quantity fields will be set to zero.

<i>Table 6-3. MDSETOTALS - Transaction Merchndise Totals Amount</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
FuncType	smallint	The code indicating the type of transaction function. See Table H-2 on page H-3 for a list of valid values for this field.
TaxblSales	decimal(15,0)	The total amount of merchandise sales that is taxable. This value is the totals sales minus and merchandise discounts for the transaction.
NTaxblSales	decimal(15,0)	The total amount of merchandise sales that is non-taxable. This value is the totals sales minus and merchandise discounts for the transaction.
TaxblRefund	decimal(15,0)	The total amount of merchandise refunds that is taxable.
NTaxblRefund	decimal(15,0)	The total amount of merchandise refunds that is non-taxable.
TaxblDiscount	decimal(15,0)	The total amount of transaction discounts for sales that are taxable.
NTaxblDiscount	decimal(15,0)	The total amount of transaction discounts for sales that are non-taxable.
TaxblFees	decimal(15,0)	The total amount of transaction fees that is taxable.
NTaxblFees	decimal(15,0)	The total amount of transaction fees that is non-taxable.
Hourly	integer	The hour which the items were sold.

TAXAMT - Transaction Tax

This view gives a summary of the tax information.

The summary excludes post voids, voids, cancels, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

Note: Taxes that have been calculated using a taxcode are not reflected in this view. To get totals transaction taxes, the TAXCODE view should be used in addition to the TAXAMT view.

<i>Table 6-4. TAXAMT - Amount of Tax on Sold Items</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
FuncType	integer	The transaction function type. See "Valid Types of Transaction Functions" on page H-3 for descriptions of the function types.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
TaxType	smallint	The type of tax collected for the transaction.
TAX	decimal(15,0)	The total amount of tax for this merchandise sale. The tax amount will be negative for all Refunded taxes. This includes returned taxes and VPLI voided taxes.
VatTax	numeric(1,0)	Flag indicating this is a Value added tax.
LuxTax	numeric(1,0)	Flag indicating this is a Luxury Tax.
Hourly	integer	The hour which the items were sold.

TAXCODE - Transaction Tax Tax Code

This view gives a summary of the tax code information for manual and automatic tax codes.

The summary excludes post voids, voids, cancels, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

<i>Table 6-5. TAXCODE - Transaction Tax Tax Code</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
FuncType	integer	The transaction function type. See "Valid Types of Transaction Functions" on page H-3 for descriptions of the function types.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
TaxType	smallint	The type of tax collected for the transaction.
TaxCode	smallint	The tax code/group number associated with this entry.
TaxTbl	smallint	The tax table associated with the tax code.
Tax	decimal(15,0)	The total amount of tax for this merchandise sale. The tax amount will be negative for all Refunded taxes.
Refunded	numeric(1,0)	Flag indicating this is a refunded tax.
Hourly	integer	The hour which the items were sold.

DISCOUNT - Transaction Discounts

This view gives a summary of the discounts given for a whole transaction by each cashier and register.

The summary excludes post voids, voids, cancels, voided layaways, voided discounts, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

Note: This table contains the discounts applied to the whole transaction. There could also be item discounts that were given for this transaction as well. These discounts can be obtained from the ITEMDISCOUNT table.

<i>Table 6-6. DISCOUNT - Transaction Discounts</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
DiscType	smallint	Value indicating the type of discount. The valid values are: 1 = default transaction discount 2 = item discount 3 = employee discount 4 = club 1 discount 5 = club 2 discount 6 = automatic discount 7 = preferred customer 9 = Other
Reason	char (6)	The reason for the transaction discount. SA <div style="border: 1px solid black; padding: 2px; display: inline-block;">For Supermarket Application This is the discount Group number.</div>
AmtDisc	decimal(15,0)	The total amount of the discount applied to the transaction.
Returned	numeric(1,0)	Returned discount flag. This flag is set to 1 for all Returned discounts.
Hourly	integer	The hour which the items were sold.

TENDER - Transaction Tender Amounts

This view gives a summary of the tenders issued for each transaction.

The summary excludes post voids, voids, cancels, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

<i>Table 6-7. TENDER - Transaction tender amounts</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
AmtTend	decimal(15,0)	The total amount tendered for this transaction for the specified tender type. This is the domestic value of the currency.
AmtForeign	decimal(15,0)	The total amount of foreign currency tendered for this transaction for the specified tender type.
AmtTendBas	decimal(15,0)	The total amount of base currency tendered for this transaction for the specified tender type. This is the Primary Base conversion amount of the Foreign currency.
AmtDue	decimal(15,0)	The balance due for this account.
TndrType	smallint	The code indicating the type of tender. For a complete list of tender types refer to Table H-1 on page H-1
ExchRate	integer	The code indicating the exchange rate of the foreign currency. Note: If the currency is not a foreign currency, this may be NULL.
ISOCurNum	integer	The currency code indicating the foreign currency. Note: If the currency is not a foreign currency, this may be NULL.
Hourly	integer	The hour which the items were sold.

TENDERCHG - Amount of Tender Change

This view gives a summary of the tender change for each cashier and register. This is the change issued for a gift certificate tender or when money is credited for a return.

The summary excludes post voids, voids, cancels, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

<i>Table 6-8. TENDERCHG - Amount of tender change</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
AmtTendChg	decimal(15,0)	The total amount of tender change for this transaction for the given tender type.
TndrType	smallint	The code indicating the type of tender. For a complete list of tender types refer to Table H-1 on page H-1
Hourly	integer	The hour which the items were sold.

TRANSVOID - Sales on Void Transaction

This view gives a summary of the cancelled items. This view excludes post voids, and training mode operations.

Note: The transvoid view does not include sales tax for the amount voided.

<i>Table 6-9. TRANSVOID - Amount of Sales Lost to Cancels</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database.
TranNum	integer	Transaction number as it appears from the terminal.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal.
PosVoided	decimal(15,0)	The total positive amount of merchandise sales from a canceled/voided transaction
NegVoided	decimal(15,0)	The total negative amount of merchandise sales from a canceled/voided transaction
Hourly	integer	The hour which the items were sold.

POSTVOID - Sales on Post Void Transactions

This view gives a summary of the post voided items.

GSA

For **General Sales Application** postvoided transaction will check the following data with the original transaction. If this data is not correctly inputted Retail Application Logical Interface will not recognize this as a post voided transaction.

Physical Store Number

Transaction Number

Terminal Number

Transaction totals must equal voided total amount

SA

For **Supermarket Application** This is not supported.

Table 6-10. POSTVOID - Amount of Sales Lost to Post Voids

Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where the original transaction was entered into the database.
TranNum	integer	Original transaction number as it appears from the terminal.
TDate	date	Original date as it appears in the transaction log header record.
TTime	time	Original time as it appears in the transaction log header record.
OperNum	char (10)	Original operator number of the person using the terminal.
AmtPVoid	decimal(15,0)	The total amount of merchandise sales that were post voided.
Hourly	integer	The hour which the items were sold.

DISCREPANCY - Transaction Discrepancy Table

This view gives a list of the potential transaction discrepancies for post voided transactions, VPLI transactions, standalone transactions, and suspended transactions. A discrepancy entry for these transactions will exist for the following conditions:

Post voided(1009)

For GSA postvoided transaction will check the following data with the original transaction. If this data is not correctly inputted Retail Application Logical Interface will not recognize this as a post voided transaction.

- Physical Store Number
- Transaction Number
- Terminal Number
- Transaction totals must equal voided total amount

Void previous line item(1010)

For GSA postvoided transaction will check the following data with the original transaction. If this data is not correctly inputted Retail Application Logical Interface will not recognize this as a post voided transaction.

- Physical Store Number
- Transaction Number
- Terminal Number
- Item number of voided item

Standalone(4004)

Standalone entries will be checked to see that the offline entries are re-entered.

— GSA —

For **General Sales Application** just the first and last offline transaction will be checked. The following data will be checked with the original transaction. If this data is not correctly inputted Retail Application Logical Interface will not recognize this as a post voided transaction.

- Physical Store Number
- Transaction Number
- Terminal Number
- Operator Number
- Transaction Date

Suspended(4010)

Suspended transactions will be check to see that this has been resumed.

— GSA —

For **General Sales Application** suspended transactions will check the following data with the original transaction. If this data is not correctly inputted Retail Application Logical Interface will not recognize this as a post voided transaction.

- Physical Store Number
- Transaction Number
- Terminal Number
- Operator Number
- Transaction Date
- Transaction Time

For **Supermarket Application** This is not supported.

Table 6-11. DISCREPANCY - Transaction Discrepancy Table

Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database.
TranNum	integer	Transaction number as it appears from the terminal.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal.
FuncType	integer	The transaction function type. See "Valid Types of Transaction Functions" on page H-3 for descriptions of the function types.
OrgWrkStaNum	char (4)	The number identifying the physical device where the original transaction was entered into the database.
OrgTranNum	integer	Original transaction number as it appears from the terminal.
OrgOperNum	char (10)	Original operator number of the person using the terminal.
ItemNum	char (26)	The item number for VPLI void transaction.

MDSESALES - Merchandise Sales Table

This view gives a summary of the merchandise sales for each cashier and register. The summary includes all discounts and allowances per each item. Both merchandise and non-merchandise sales may be viewed, as specified by the NonMerch flag.

The summary excludes post voids, voids, cancels, layaways, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

It does not remove transaction discounts.

<i>Table 6-12. MDSESALES - Merchandise Sales Table</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
DeptNum	char (6)	The department associated with the sale of this item.
Class	char (6)	The class number associated with the sale of this item.
ItemNum	char (26)	The item number.
AmtSales	decimal(15,0)	The total amount of merchandise sales for this item number in the class and department.
AmtDisc	decimal(15,0)	The total amount of merchandise discounts for this item number in the class and department. This includes all item discounts and allowances
QtySales	decimal(10,2)	The total number of items sold for this item number in the class and department. Or the total weight of the item sold. See the Weightltm flag below.
Returned	numeric(1,0)	Returned Merchandise totals flag. This flag is set to 1 for all Returned Merchandise.
NonMerch	numeric(1,0)	Non-Merchandise flag. This flag is set to 1 for all non-merchandise items.
Weightltm	numeric(1,0)	Weight item flag. This flag is set to 1 for all items requiring a scale weight.
Reason	char (6)	The reason of the non-merchandise sale.
Hourly	integer	The hour which the items were sold.

LAYMDSE - Layaway Merchandise Sales Table

This view gives a summary of the layaway merchandise sales for each cashier and register. This includes both merchandise and non-merchandise sales, as specified by the NonMerch flag.

The summary excludes post voids, voids, cancels, voided layaways, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

SA

For **Supermarket Application** This is not supported.

Table 6-13. LAYMDSE - Layaway Merchandise Sales Table

Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
DeptNum	char (6)	The department associated with the sale of this item.
Class	char (6)	The class number associated with the sale of this item.
ItemNum	char (26)	The item number.
AmtSales	decimal(15,0)	The total amount of merchandise sales for this item number in the class and department.
QtySales	decimal(10,2)	The total number of items sold for this item number in the class and department.
Returned	numeric(1,0)	Returned Merchandise totals flag. This flag is set to 1 for all Returned Merchandise.
NonMerch	numeric(1,0)	Non-Merchandise flag. This flag is set to 1 for all non-merchandise items.
Reason	char (6)	The reason of the non-merchandise sale.
Hourly	integer	The hour which the items were sold.

ITEMDISCOUNT - Item Discounts Given

This view gives a summary of the discounts given on items by each cashier and register.

The summary excludes post voids, voids, cancels, voided layaways, voided discounts, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

Note: This table contains the discounts applied to the individual items. There could also be transaction discounts given for this transaction as well.

SA

For **Supermarket Application** This is not supported.

Table 6-14. ITEMDISCOUNT - Item Discounts Given

Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
ItemNum	char (26)	Item number the discount was applied to.
SeqNum	smallint	The sequence number to match keys for item discounts. Note: This would be to match items from the SPRNSALES table with the items here to get an accurate discount snapshot.
DiscType	smallint	Value indicating the type of discount. The valid values are: 1 = default transaction discount 2 = item discount 3 = employee discount 4 = club 1 discount 5 = club 2 discount 6 = automatic discount 7 = preferred customer 9 = Other
Reason	char (6)	The reason for the item discount.
AmtDisc	decimal(15,0)	The total amount of the discount applied to the transaction.
Returned	numeric(1,0)	Returned discount flag. This flag is set to 1 for all Returned discounts.
Hourly	integer	The hour which the items were sold.

ITEMDEPOSIT - Item Deposits

This view gives a summary of the deposits on items by each cashier and register.

The summary excludes post voids, voids, cancels, voided layaways, voided discounts, training mode transactions, standalone(offline) transactions, and suspended transactions.

Note: This table contains the deposits applied to the individual Items.

GSA

For **General Sales Application** This is not supported.

Table 6-15. ITEMDEPOSIT - Item Deposits

Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
ItemNum	char (26)	Item number for the deposit.
Amount	decimal(15,0)	The total amount of the deposit for the item.
Returned	numeric(1,0)	Returned discount flag. This flag is set to 1 for all Returned deposits.
Hourly	integer	The hour which the items were sold.

ITEMVOID - Sales on Voided Items

This view gives a summary of the voided items. This view excludes post voids, layaways, returns, void of previous line items, and training mode operations.

Note: The itemvoid view does not include sales tax for the amount voided.

<i>Table 6-16. ITEMVOID - Amount of Sales Lost to Item Voids</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database.
TranNum	integer	Transaction number as it appears from the terminal.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal.
DeptNum	char (6)	The department associated with the sale of this item.
Class	char (6)	The class number associated with the sale of this item.
ItemNum	char (26)	The item number.
AmtVoid	decimal(15,0)	The total amount of merchandise sales that are voided items for this item number for the department number and class.
DscVoid	decimal(15,0)	The total discounted amount of merchandise sales that are voided items for this item number for the department number and class.
QtyVoid	decimal(10,2)	The total number of items voided for this item number for the department number and class.
Hourly	integer	The hour which the items were sold.

VPLIVOID - Sales of Line Item Voids

This view gives a summary of the item sales lost to line item voids. This reflects the original item entries that were voided.

SA

For **Supermarket Application** This is not supported.

<i>Table 6-17. TVPLIVOID - Amount of Sales Lost to Line Item Voids</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The Original number identifying the physical device where the original transaction was entered into the database.
TranNum	integer	Original Transaction number as it appears from the terminal.
TDate	date	Date as it appears in the transaction log header record. <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> <p style="text-align: center;">GSA</p> <p>The General Sales Application VPLI function does not set the original transaction date and time. The VPLI header date and time will be used. A VPLI issued on a different day than the original transaction will not be reflected in these views.</p> </div>
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person who inputted the original transaction.
DeptNum	char (6)	The department associated with the sale of this item.
Class	char (6)	The class number associated with the sale of this item.
ItemNum	char (26)	The VPLI item number.
NonMerch	numeric(1,0)	Flag indicating this is a non-merchandise item.
Returned	numeric(1,0)	Flag indicating this is a returned item.
Txblamt	decimal(15,0)	The VPLI amount of item sales that is taxable.
NTxblamt	decimal(15,0)	The VPLI amount of item sales that is non-taxable.
Txbldisc	decimal(15,0)	The VPLI discount amount of item sales that is taxable.
NTxbldisc	decimal(15,0)	The VPLI discount amount of merchandise sales that is non-taxable.
Quantity	decimal(10,2)	The total number of items voided for this item number for the department number and class.
Hourly	integer	The hour which the items were sold.

ISSUETNDR - Gift Certificate and Store Credit Issued

This view gives a summary of the gift certificates and store credit amounts issued by each cashier and register.

The summary excludes post voids, voids, cancels, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

it also excludes redeemed store credit and gift certificates, and voided store credit and gift certificates.

<i>Table 6-18. ISSUETNDR - Gift Certificate and Store Credit Issued</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
AmtTender	decimal(15,0)	The total amount of gift certificate or store credit issued.
TndrType	smallint	The code indicating the type of tender. For a complete list of tender types refer to Table H-1 on page H-1
Hourly	integer	The hour which the items were sold.

ACCTNEW - Total Amount of New/Adjusted Accounts

This view gives a summary of the new/adjusted accounts created for each cashier and register.

The summary excludes post voids, voids, cancels, voided layaways, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

Note: The amounts in the new layaway table include any fees associated with a new layaway.

SA

For **Supermarket Application** This is not supported.

<i>Table 6-19. ACCTNEW - Total Amount of New/Adjusted Accounts</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
AcctNum	char (24)	The account number for this layaway.
AcctType	smallint	The type of account associated with the payment. The valid type of accounts; 1 = Customer account 2 = COD 3 = Layaway 4 = Job 9 = Other
Amount	decimal(15,0)	The total amount of new/adjusted account. This is the amount that is charged to the account.
Adjust	numeric(1,0)	Account adjustment flag. This flag is set to 1 for all entries that are adjustments to the account.
Hourly	integer	The hour which the items were sold.

ACCTPAY - Total Amount of Account Payments

This view gives a summary of the payments made for each cashier and register.

The summary excludes post voids, voids, cancels, voided layaways, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

SA

For **Supermarket Application** This reflects the payments for a Miscellaneous item type(4,5).

Table 6-20. ACCTPAY - Total Amount of Account Payments

Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
AcctType	smallint	The type of account associated with the payment. The valid type of accounts; 1 = Customer account 2 = COD 3 = Layaway 4 = Job 9 = Other
AcctNum	char (24)	The account number for this payment.
Amount	decimal(15,0)	The total amount of account payments made.
Returned	numeric(1,0)	Returned payment flag. This flag is set to 1 for all Returned payments due to a canceled account or merchandise item.
Hourly	integer	The hour which the items were sold.

ACCTDEP - Total Amount of Account Deposits

This view gives a summary of the deposits made for each cashier and register.

The summary excludes post voids, voids, cancels, voided layaways, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

SA

For **Supermarket Application** This is not supported.

<i>Table 6-21. ACCTDEP - Total Amount of Account Deposits</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
AcctType	smallint	The type of account associated with the payment. The valid type of accounts; 1 = Customer account 2 = COD 3 = Layaway 4 = Job 9 = Other
AcctNum	char (24)	The account number for this deposit.
Amount	decimal(15,0)	The total amount of account deposits made.
Returned	numeric(1,0)	Returned payment flag. This flag is set to 1 for all returned deposits due to a canceled account or merchandise item.
Hourly	integer	The hour which the items were sold.

ACCTFEE - Total Amount of Fees on Accounts

This view gives the summary of the fees charged on accounts for each cashier and register for fees made.

The summary excludes post voids, voids, cancels, voided layaways, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

SA

For **Supermarket Application** Only tender handling fee is supported.

Table 6-22. ACCTFEE - Total Amount Account Fees

Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
AcctNum	char (24)	The account number for this fee.
FeeType	decimal(15,0)	The type of fee. The valid values are: 1 = Send 2 = COD 3 = Layaway 4 = Tender Handling 5 = Restocking 9 = Other
Amount	decimal(15,0)	The total amount of account fees.
Returned	smallint	Flag to indicated if this fee amount was returned.
Hourly	integer	The hour which the items were sold.

LAYCAN - Total Amount of Cancels on Layaways

This view gives the summary of layaway cancels made on accounts for each cashier and register. This reflects the total amount returned for the layaway.

The summary excludes post voids, voids, cancels, voided layaways, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

SA

For **Supermarket Application** This is not supported.

Table 6-23. LAYCAN - Total Amount of Cancels on Layaways

Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
AcctNum	char (24)	The account number for this cancel.
Amount	decimal(15,0)	The total amount of money returned because of a cancelled layaways. This includes the payment and deposit amounts, but does not include any fees returned.
Hourly	integer	The hour which the items were sold.

LAYRSTK - Total Amount of Restocks on Layaways

This view gives the summary of layaway restocks made on accounts for each cashier and register.

The summary excludes post voids, voids, cancels, voided layaways, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

SA

For **Supermarket Application** This is not supported.

Table 6-24. LAYRSTK - Total Amount of Restocks on Layaways

Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
AcctNum	char (24)	The account number for this cancel.
Amount	decimal(15,0)	The total amount of money from a restock of merchandise because of cancelled layaways. This only reflects the original amount when the layaway was initiated.
Hourly	integer	The hour which the items were sold.

PICKUPS - Transaction Pickup Amounts

This view gives a summary of the pickups made for each cashier and register.

The summary excludes post voids, voids, cancels, layaways, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

SA

For **Supermarket Application** This is tenders picked up for transaction type 4.

GSA

For **General Sales Application** This is tenders picked up for transaction type 51 and 63.

<i>Table 6-25. PICKUPS - Transaction pickup amounts</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
Amount	decimal(15,0)	The total amount picked up for this transaction for the specified tender type.
TndrType	smallint	The code indicating the type of tender. For a complete list of tender types refer to Table H-1 on page H-1
Hourly	integer	The hour which the items were sold.

PAYOUTS - Transaction Payout Amounts

This view gives a summary of the payouts made for each cashier and register. This is the amount of money withdrawn from the register.

The summary excludes post voids, voids, cancels, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

SA

For **Supermarket Application** This is not supported.

GSA

For **General Sales Application** This is not supported.

Table 6-26. PAYOUTS - Transaction payout amounts

Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
Reason	char (6)	Reason for the payout.
Amount	decimal(15,0)	The total amount payed out for this transaction for the specified tender type.
TndrType	smallint	The code indicating the type of tender. For a complete list of tender types refer to Table H-1 on page H-1
Hourly	integer	The hour which the items were sold.

PAYINS - Transaction Payin Amounts

This view gives a summary of the payins(Loans) made for each cashier and register. This is the amount of money deposited into the register.

The summary excludes post voids, voids, cancels, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

SA

For **Supermarket Application** This is tenders loaned for transaction type 3.

GSA

For **General Sales Application** This is tenders loaned for transaction type 50.

<i>Table 6-27. PAYINS - Transaction payin amounts</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
Reason	char (6)	Reason for the payin.
Amount	decimal(15,0)	The total amount payed in for this transaction for the specified tender type.
TndrType	smallint	The code indicating the type of tender. For a complete list of tender types refer to Table H-1 on page H-1
Hourly	integer	The hour which the items were sold.

TNDRCASH - Tender Cashed Amounts

This view gives a summary of the tenders cashed for each cashier and register.

The summary excludes post voids, voids, cancels, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

GSA

For **General Sales Application** This is not supported.

Table 6-28. TNDRCASH - Tender cashed amounts

Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
Amount	decimal(15,0)	The total amount of the tenders cashed.
TndrType	smallint	The code indicating the type of tender being cashed. For a complete list of tender types refer to Table H-1 on page H-1
Hourly	integer	The hour which the items were sold.

SPRNSALES - Total Amount of Salesperson Sales

This view gives the summary of salesperson sales made by each salesperson.

The summary excludes post voids, voids, cancels, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

<i>Table 6-29. SPRNSALES - Total Amount of Salesperson Sales</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
SalesPrsn	char (10)	<p>ID of the person who sold the item.</p> <p>— GSA —</p> <p>General Sales Application doesn't provide a salespersn, so this will be set to the transaction operator.</p> <p>— SA —</p> <p>Supermarket Application doesn't provide a salespersn, so this will be set to the transaction operator.</p>
DeptNum	char (6)	The department associated with the sale of this item.
Class	char (6)	The class number associated with the sale of this item.
ItemNum	char (26)	The item number.
Returned	smallint	Flag to indicated if this was a return of a salesperson sale.
AmtSales	decimal(15,0)	The total amount of salesperson sales.
QtySales	decimal(10,2)	The total number of items sold for this item number in the class and department.
Hourly	integer	The hour which the items were sold.

SPECIALSALES - Special Sales Amounts

This view gives a summary of the special sales made. These special sales are for the sales amounts where discounts were given in the ITEMDISC table. The most common special sale would be the employee sales (DISCTYPE = 3) and preferred customer (DISCTYPE = 7). Other sales information is also included.

The summary excludes post voids, voids, cancels, training mode transactions, standalone(offline) transactions, and suspended transactions. Standalone transactions will be included in this view when the corresponding transactions are re-entered. Suspended transactions will be included when the transaction is resumed.

Note: This table contains the sales amounts where a discount was given. These amounts are included in the MDSESALES view and are in this view to enable breakdown of data for employee and preferred customer sales.

<i>Table 6-30. SPECIALSALES - Special Sales Made</i>		
Name	Data type	Description
PhyStorNum	char (6)	Physical Store number. This shows the Physical Store number of the store which the totals are from.
WrkStaNum	char (4)	The number identifying the physical device where this transaction was entered into the database. For re-entered transactions, the workstation number will be set to the original workstation of the offline transaction.
TranNum	integer	Transaction number as it appears from the terminal. For re-entered transactions, the transaction will be set to the original transaction number of the offline transaction.
TDate	date	Date as it appears in the transaction log header record.
TTime	time	Time as it appears in the transaction log header record.
OperNum	char (10)	Operator number of the person using the terminal. For re-entered transactions, the operator will be set to the original operator of the offline transaction.
DeptNum	char (6)	The department associated with the sale of this item.
Class	char (6)	The class number associated with the sale of this item.
ItemNum	char (26)	Item number the discount was applied to.
AmtSales	decimal(15,0)	The total amount of special sales for this item number in the class and department.
QtySales	decimal(10,2)	The total number of items sold for this item number in the class and department.
SaleType	smallint	Value indicating the type of discount. The valid values are: 1 = default transaction sale 2 = item sale 3 = employee sale 4 = club 1 sale 5 = club 2 sale 7 = preferred customer sale 9 = Other sale
AcctNum	char (24)	The account number for this sale. Note: This will be the employee number if the SaleType = 3 and the preferred customer number when the SaleType = 6.
Hourly	integer	The hour which the items were sold.

Chapter 7. RAIS Applications/Reports

This chapter gives you the information needed to work with the RAIS Retail Application Logical Interface(RALI) Applications/Reports option (option 50).

The RAIS applications manipulate the RAIS database collection and reports can be generated using the data stored in a RAIS database collection. The reports are generated using the Retail Application Logical Interface(RALI) which is discussed in the Chapter 6, "Retail Application Logical Interface" on page 6-1. The applications and reports are accessed from Option 4 of the Retail Application Interface Suite Main Menu or through a CL command. Selecting this option will bring up the *RAIS Applications/Reports* menu.

```
SR AIS          Retail Logical Interface Applications
                                     System:  RCHASHTL
Select one of the following:

  1. Delete transactions
  2. Print hourly totals report
  3. Print cumulative item sales totals
  4. Print item sales hourly count
  5. Print transactions hourly count
  6. Print cashier hourly totals
  7. Print register hourly totals
  8. Print hourly productivity totals
  9. Print sales person totals
 10. Print sales person department totals

Selection or command
====>

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel

(C) COPYRIGHT IBM CORP.
```

Note: One of the Product options for RAIS must be installed for option 4 to be shown off the main menu. Also the applications will not run unless one is installed.

Application/Report Command Definitions

The Menu options will bring up the CL command interface to the delete application and reports. The CL Command for the Remove Transactions menu option is the Remove Retail Transactions (RMVRTLTNS) command. The CL command for the reports is Print Retail Report (PRTRTLRPT) command. This command will generate all reports that are defined in this chapter.

A syntax diagram follows for both of the commands. The diagrams show required and optional (defaulted) parameters for each command.

Remove Retail Transactions (RMVRTLTNS) Command

This application allows the removal of a group of transactions that are dated:

- on or before a specified number of days
- on or before some specified date

This is a useful tool to keep the database size under control.

Note: If the RAIS database is the only collection of the Retail Application TLOG data, you should first save off the Transactions before deleting them.

```
Job: B,I Pgm: B,I REXX: B,I Exec
Threadsaf e: No

RMVRTLTNS (P)AGE( number-days-old )
          DATE( remove-date )

Note:
P All parameters preceding this point can be specified positionally.
```

AGE Parameter

This parameter is used to specify that all transactions that are dated on or before the current date minus this number of days will be removed from the relational database collection.

number-days-old Only Retail Application Logical Interface entries dated on or before the current date minus the number of days old will be removed.

DATE Parameter

This parameter is used to specify that all transactions that are dated on or before this date will be removed from the relational database collection.

remove-date Only Retail Application Logical Interface entries dated on or before the remove date will be deleted.

Note: DATE and AGE are mutually exclusive.

Print Retail Report (PRTRTLRPT) Command

This command starts up the reporting function for the Retail Application Logical Interface. The application will query the RAIS relational database collection and generate reports based on the command parameters. See "Report Definitions" on page 7-3 for definitions of the report information generated.

Note: The RAIS data library must exist in the library list for this application to generate and report data. See the "Create Retail Conversion Template" on page 2-9 for more information about the Data Library. Also, one of the optional product loads must be installed.

```
Job: B,I Pgm: B,I REXX: B,I Exec
Threadsaf: No

PRTRTLRPTTYPE( HT ) (P)
                CIST
                ISHC STORENBR( ALL
                                store-number )
                THC
                CHT
                RHT
                HPT
                SPT
                SPDT

STRDATE( TODAY
          start-date ) ENDDATE( TODAY
                                end-date ) CASHIER( ALL
                                                    cashier-number )

REGISTER( ALL
          register-number ) SALESPRSN( ALL
                                       sales-person )
```

Note:
P All parameters preceding this point can be specified positionally.

RPTTYPE Parameter

One of the following may be specified:

- *HT** The Hourly Totals report will be generated. See the "Report Definitions" on page 7-3 for information on this report.
- *CIST** The Cumulative Item Sales Totals report will be generated. See the "Report Definitions" on page 7-3 for information on this report.
- *ISHC** The Item Sales Hourly Count report will be generated. See the "Report Definitions" on page 7-3 for information on this report.
- *THC** The Transactions Hourly Count report will be generated. See the "Report Definitions" on page 7-3 for information on this report.
- *CHT** The Cashier Hourly Totals report will be generated. See the "Report Definitions" on page 7-3 for information on this report.

- *RHT** The Register Hourly Totals report will be generated. See the "Report Definitions" on page 7-3 for information on this report.
- *HPT** The Hourly Productivity Totals report will be generated. See the "Report Definitions" on page 7-3 for information on this report.
- *SPT** The Sales Person Totals report will be generated. See the "Report Definitions" on page 7-3 for information on this report.
- *SPDT** The Sales Person Department Totals report will be generated. See the "Report Definitions" on page 7-3 for information on this report.

STORENBR Parameter

This parameter specifies the Store Number to use when generating the report. This is compared to the Retail Application Logical Interface PhyStorNum field.

***ALL** All store entries will be use to generate the report.

store_number Only Retail Application Logical Interface entries with this store number will be used to generate the report.

STRDATE Parameter

This parameter specifies the starting date to use when generating the report. This is compared to the Retail Application Logical Interface TDate field.

***TODAY** Only Retail Application Logical Interface entries with start dates greater than or equal to todays date will be used.

start-date Only Retail Application Logical Interface entries with start dates greater than or equal to this configured date will be used.

ENDDATE Parameter

This parameter specifies the ending date to use when generating the report. This is compared to the Retail Application Logical Interface TDate field.

***TODAY** Only Retail Application Logical Interface entries with end dates less than or equal to todays date will be used.

start-date Only Retail Application Logical Interface entries with end dates less than or equal to this configured date will be used.

CASHIER Parameter

This parameter specifies the cashier entries to use when generating the report. This is compared to the Retail Application Logical Interface OperNum field for the *CHT report type.

***ALL** All Retail Application Logical Interface OperNum entries will be used.

selected-cashier-for-report Only Retail Application Logical Interface entries with comparable selected-entries will be used.

REGISTER Parameter

This parameter specifies the register entries to use when generating the report. This is compared to the Retail Application Logical Interface WrkStaNum field for the *RHT report type.

***ALL** All Retail Application Logical Interface WrkStaNum will be used.

selected-register-for-report Only Retail Application Logical Interface entries with comparable selected-entries will be used.

SALESPRSN Parameter

This parameter specifies the sales person entries to use when generating the report. This is compared to the Retail Application Logical Interface SalesPrsn field for the *SPT and *SPDT report types.

***ALL** All Retail Application Logical Interface SalesPrsn entries will be used.

selected-sales-person-for-report Only Retail Application Logical Interface entries with comparable selected-entries will be used.

Report Definitions

Hourly Totals Report

The Hourly Totals report shows hourly sales statistics for a store. This report gives a total sales per hour and a daily total for each department.

This application can also be executed using the **Print Hourly Totals report** (*HT specified for the RPTTYPE parameter on the PRTRTLRPT command)

For each department, the report presents the following:

DEPARTMENT	The reporting department.
DATE	The date of day being reported on.
HOUR	The ordinal hour of day being reported on.
AMOUNT	The total amount of merchndise and non-merchandise sales in the hour, based on the day's total.

Cumulative Item Sales Totals Report

The cumulative item sales totals report shows the top ten merchandise item sales totals. The report gives item sales totals in two parts; by top dollar amount, and by top quantity sold.

This application can also be executed using the **Print Cumulative Item Sales Totals report** (*CIST specified for the RPTTYPE parameter on the PRTRTLRPT command).

For a specified store, the report presents the following information for the top ten items by top dollar amount and then the top ten items by quantity sold:

ITEM	The merchandise item number of the top ten item.
# SOLD	The number of merchandise items sold.
\$ VALUE	The total amount of merchandise sales of the item.

Item Sales Hourly Count Report

The items sales hourly count report shows hourly count of merchandise items sold each day for each work station.

This application can also be executed using the **Print Item Sales Hourly Count report** (*ISHC specified for the RPTTYPE parameter on the PRTRTLRPT command).

For each workstation, the report presents the following:

WORKSTATION	The reporting workstation.
DATE	The date of day being reported on.
HOUR	The ordinal hour of day being reported on.
# OF ITEMS	The total number of merchandise items sold in the hour for this workstation.

Transactions Hourly Count Report

The transactions hourly count report shows hourly count of transactions each day for each work station.

This application can also be executed using the **Print Transactions Hourly Count report** (*THC specified for the RPTTYPE parameter on the PRTRTLRPT command).

For each workstation, the report presents the following:

WORKSTATION	The reporting workstation.
DATE	The date of day being reported on.
HOUR	The ordinal hour of day being reported on.
# OF TRANSACTIONS	The total number of transactions in the hour for this workstation.

Cashier/Register Hourly Totals Reports

The cashier/register hourly totals reports are summaries of the tender status and the statistics for sales. Each report prints the summaries for each cashier/register and then prints the summary for the entire store. The tender status portion gives the summary of the tender type, the amount of the tender coming in, the amount of the tender going out, and the net amount of tender. The statistics portion of the report gives the summary of merchandise sales, discount information, layaway information and miscellaneous information.

These applications can also be executed using the **Print Cashier report** (*CHT specified for the RPTTYPE parameter on the PRTRTLRPT command) and the **Print Register report** (*RHT specified for the RPTTYPE parameter on the PRTRTLRPT command).

The tender status portion of the report presents the following:

MEDIA	The type of tender used as payment (such as cash, checks, visa).
IN	The total dollar amount of tender put into the register.
OUT	The total dollar amount of tender taken out of the register.
NET	The difference between the tender put into the register and the tender taken out.
TENDER RECONCILIATION	The total amount of tender in the drawer. This should equal the net totals. Foreign currency is shown with the converted value to be in the current currency.

INFLOW/OUTFLOW ANALYSIS

The transactions for which tender is put into and taken out of the register.

NET IN/OUT

The summation of the inflow/outflow analysis.

The statistics portion of the report presents the following:

SALES INFO

The total amount of merchandise sales and returns, gift certificate, employee, non-merchandise, taxable, and non-taxable sales along with the total amount of sales, luxury and value added tax.

SALES STATISTICS

The number of sales and items sold, the average number of units sold in a sale, the average dollar amount made in a sale, the number of returns and items returned, the average number of units returned, and the average dollar amount in a return.

MISCELLANEOUS INFO

The total number of cancelled transactions, the dollar amount of the cancels excluding tax, the total number of post voids made, the total dollar amount of the post voids excluding tax, the total number of voids made, the total dollar amount of the voids, and the number of non-sale transactions.

DISCOUNT INFO	The total dollar amount of merchandise and employee discounts given.
LAYAWAY INFO	The total amount of layaway fees, the total dollar amount of layaway sales, the total dollar amount of adjustments on layaway, the total dollar amount of cancel-restock layaways, the total dollar amount of payments on layaways, the total dollar amount of restock layaway deposits, the total dollar amount of sales tax on layaways, the total dollar amount of luxury tax on layaways, and the total dollar amount of value added tax on layaways.
IN/OUT INFO	The total dollar amount of on-account payments, the number of cash payouts made, the total dollar amount of cash payouts, the number of payins, the total dollar amount of payins, the number of pickups, and the total dollar amount of pickups.
SALES RECONCILIATION	The total dollar amount of merchandise sales made, the total dollar amount of returns, the net sales(merchandise sales less returns), the total dollar amount of taxable sales, the total dollar amount of non-taxable sales, the total dollar amount of layaway fees, and the net sales total(sales totals less the layaway fees).

Hourly Productivity Report

The hourly productivity report shows hourly productivity for a store. The report gives a total number of customers, units, sales, average units, average customers, % of customers per hour and % of sales per hour.

This application can also be executed using the **Print Hourly Productivity Totals report** (*HPT specified for the RPTTYPE parameter on the PRTRTLRPT command).

For each hour, the report presents the following:

HOUR	The ordinal hour of day being reported on.
#CUST	The total number of customers served in the hour.
NET UNITS	The total number of merchandise items sold in the hour.
NET ACTIVITY	The total dollar amount of merchandise sales in the hour.
AVG UNITS	The average number of units sold in the hour.
AVG \$\$\$\$\$	The average dollar amount spent on an item in the hour.
%CUST	The percent of customers in the hour, based on the day's total.
\$\$\$\$\$	The percent of amount of sales in the hour, based on the day's total.

Salesperson Totals Report

The salesperson totals report shows each salesperson's total sales and returns. The report also gives a total amount of sales and returns for all salespeople. The report may be run for one or all salespeople and for one date or a range of dates per store

Included in the salesperson report are merchandise and non-merchandise sales and returns. Excluded from the report are voids.

This application can also be executed using the **Print Salesperson Totals report** (*SPT specified for the RPTTYPE parameter on the PRTRTLRPT command).

For each salesperson, the report presents the following:

SPRSN #	The salesperson's number.
SALES	The total dollar amount of items sold, excluding voids and returns.
RETURNS	The total dollar amount of items returned, excluding voids.
ITEMS SOLD	The total number of items sold, excluding voids and returns.
ITEMS RETURNED	The total number of items sold excluding voids.

Salesperson Department Totals Report

The salesperson department totals report shows the salesperson total sales from each class in each department, as well as the salesperson total sales from each department. The report also presents the total number of items sold and sales by all salespeople for all departments. This report can assist you with commission calculation.

This application can also be executed using the **Print Salesperson Department Totals report** (*SPDT specified for the RPTTYPE parameter on the PRTRTLRPT command).

The report presents the following for each salesperson:

SPSN The salesperson number.
NAME The salesperson name.

DEPT

The department the sold items are in.

CLASS

The class the sold items are in. There may be one or more classes within the department.

UNITS

The total number of items sold.

SALES

The total dollar amount of sales.

MKDNS

The total dollar amount of markdowns (difference between current price and the original price).

Appendix A. Problem Analysis

This appendix provides information and guidance for problem analysis.

Detecting the Problem

Whenever an irrecoverable error is detected, a message is issued to the console and logged in the AS/400 history log (QHST). This is often the first indication that a problem has been detected.

Certain communication functions can successfully start but not complete. To determine whether a function has not completed on the AS/400 system, you can display all active users of the system.

Certain functions, such as a pass-through session for credit authorization or electronic funds, normally run continually.

If the AS/400 system is an in-store processor running unattended and a problem is detected by the host system, you should display the AS/400 history log (QHST) to the host system to inspect the messages.

Re-creating the Problem

First, you must gather all available problem analysis information. This is done by copying or printing the AS/400 history log and using the problem handling options of the AS/400 system menus. Even when the problem can be reproduced, this information should be collected for evaluation.

If the problem can be created again with conditions that exactly match the original situation, the environment should be continually simplified until the problem is isolated or can no longer be created. This may require that you remove job steps from a procedure until the problem occurs within a single job step.

Gathering Problem Analysis Information

If you are able to duplicate the problem in its simplest form, you should do a final run with all problem analysis tools activated.

If communications are involved, run a communications trace during the final run. If the application loops, cancel the application. Job trace, QSYSARB trace, QCLUS trace, and a trace of the subsystem in which communications is running can be requested as supporting information. You should also print the history and job log of the final run.

The RAIS api and applications have an internal trace that can be turned on to get additional information on the

problem. To turn on the trace do the following before running your application.

```
ADDENVVAR ENVVAR(QIBM_ZCF_TRACE_LEVEL) VALUE(1)
```

After running the failing application, turn the trace off by issuing the following:

```
CHGENVVAR ENVVAR(QIBM_ZCF_TRACE_LEVEL) VALUE( )
```

Dump the trace with the following command:

```
DMPUSRTRC
```

This will dump the trace to the QAPOZDMP file in your QTEMP library. Include this trace in your Problem Analysis information.

If there is a problem with the polling application, trace can be started by creating the QZCFTRACE data area in QGPL. Use the following commands:

```
ENDRTL SVR
```

This will end the server. This needs to be done because tracing is only checked at startup of the polling application.

```
CRTDTAARA DTAARA(QGPL/QZCFTRACE) TYPE( CHAR)
          LEN(4) VALUE(x)
```

where x = :

'0' - RAIS internal trace will not be started.

'1' - RAIS internal trace will be started.

and where y = :

'0' - polling application job will not be traced.

'1' - polling application job will be traced.

```
STRRTL SVR
```

When polling is ended, the trace information will be dumped to the polling joblog. In addition to the trace you may want to change the polling job description message logging to log the messages.

```
CHGJOB JOB(QRAIS/QZCFPOLL) LOG(4 SECLVL)
```

Note: The polling application needs to have *USE for the TRCJOB command and the QP0WUSRT program in QSYS. The QRAIS profile should be given authority to those programs to collect the information. The QRAIS profile should also be given *ALLOBJ before running a trace to be able to dump the indexes being used internally.

Communicating the Problem

When communicating any problem to support personal, use the Support ID for this product so the problem report can get routed to the appropriate parties. The support id is **5799RAI**

Appendix B. Examples

RAIS API Example

This section contains a ILE COBOL code example showing how to use the RAIS API. For more information on ILE, see the *ILE Concepts Book*. For more information on ILE COBOL, see the *ILE COBOL for AS/400: Programmer Book*.

```
IDENTIFICATION DIVISION.

Program name: TSTILEGT

Description: Sample COBOL program reads data from
            a 469 file.

NOTES: This must be compiled using the NOMONOPRC option
       otherwise the procedure calles will be upper cased.
EX:
  CRTCBMOD MODULE( LIBL/TSTILEGT)
  SRCFILE( LIBL/QCBLLESRC)
  OPTION( NOMONOPRC) DBGVIEW( SOURCE)

  CRTPGM PGM( LIBL/TSTILEGT)
  MODULE( LIBL/TSTILEGT)
  BNDSRVPGM(QRAIS/QZCFPCMN)

PROGRAM-ID. TSTILEGT.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
  LINKAGE TYPE IS PROCEDURE FOR 'qzcfopen'
  LINKAGE TYPE IS PROCEDURE FOR 'qzcfclose'
  LINKAGE TYPE IS PROCEDURE FOR 'qzcfread'.

DATA DIVISION.
WORKING-STORAGE SECTION.

  1 HANDLE                PIC S9(9) BINARY.
  1 OPEN-FLAG             PIC S9(9) BINARY.
  1 REASON                PIC S9(9) BINARY.
  1 BUFFER                PIC X(6 ).
  1 DATA-LENGTH         PIC S9(9) BINARY.
  1 ILELocation           PIC X(32).
  1 ILEPath               PIC X(32).

  1 HEX                  PIC 9 BINARY value .
  1 NULL-VALUE           REDEFINES HEX PIC X.
  QZCFERRNO is exported from the RAIS Service program
  1 QZCFERRNO            PIC S9(9) BINARY EXTERNAL.

  Theses are 32 characters since this is driven
  by the CL CALL statement.
LINKAGE SECTION.
  1 Target-File           PIC X(32).
  1 Location              PIC X(32).

PROCEDURE DIVISION USING Target-File Location.
P .
  indicate that sample program has started
  DISPLAY "TSTILEGT start".

  The Path and Location Parameters MUST be NULL terminated.

  DISPLAY "Target " Target-File.
  DISPLAY "Location: " Location.
  string Location
  NULL-VALUE
  DELIMITED BY SPACES
  INTO ILELocation.
  string TARGET-FILE
  NULL-VALUE
  DELIMITED BY SPACES
```

```
INTO ILEPath.

Set open flag for read-only
move 1 to OPEN-FLAG.
CALL "qzcfopen"
  USING ILEPath,
  ILELocation,
  BY Value OPEN-FLAG
  RETURNING HANDLE.
IF HANDLE IS LESS THAN
  DISPLAY "unable to open TARGET errno: " QZCFERRNO
  STOP RUN
END-IF.

Read 6 bytes from the file

CALL "qzcfread"
  USING By Value HANDLE,
  By Reference BUFFER,
  By Value 6
  RETURNING DATA-LENGTH.
report reason, if any; stop if failed
IF DATA-LENGTH IS LESS THAN 1
  DISPLAY "MQREAD FAILED: " QZCFERRNO
  STOP RUN
END-IF.

DISPLAY "QZCFREAD BYTES READ: " DATA-LENGTH.

CALL "qzcfclose"
  USING By Value HANDLE
  RETURNING REASON.

STOP RUN.

END OF TSTILEGT
```

Retail Controller Configuration Examples

This section contains configuration examples for the retail controller. For more information on configuring lines, controllers, and devices, see the *Communications Configuration* book.

Nonswitched Retail Controllers Configuration

This is a sample configuration for a 4680 retail controller on a nonswitched line to the AS/400 system. Synchronous modem eliminators can be used to simulate the nonswitched line in a local environment. You may need to specify additional parameters to those shown for your communications environment.

1. Determine the AS/400 communications line resource name to use in the line description by using the Work with Hardware Resources command.

```
WRKHDWRSC TYPE( CMN)
```

2. Configure an SNA primary communications line using the Create Line Description (SDLC) command.

```
CRTLNSDLC LIND(RTLLINE) RSRNAME(LIN 22) ROLE( PRI)
EXCHID( 56 1) MAXFRAME(265)
```

3. Configure a retail communications controller using the Create Controller Description (Retail) command.

```
CRTCTLRTRL CTLD(POSC1) TYPE(468 ) MODEL( ) LINKTYPE( SDLC)
LINE(RTLLINE) EXCHID( 4D 1)
MAXFRAME(265) STNADR(C1)
```

The station address and exchange ID must match those specified when configuring the 4680 store system host communications.

4. Configure a retail device for HCP communications using the Create Device Description (Retail) command.

```
CRTDEVRTL DEVD(POSC1HCP) LOCADR( 1) RMTLOCNAME(POSC1HCP)
CTL(POSC1) MAXLENRU(256) APPTYPE( OTHER)
```

It is strongly recommended that the device name and remote location name be the same. This makes working with communications management much easier on the Work with Configuration Status system displays.

5. Configure a retail device for start user program.

```
CRTDEVRTL DEVD(POSC1SUP) LOCADR( 3) RMTLOCNAME(POSC1SUP)
CTL(POSC1) MAXLENRU(256) APPTYPE( SBMRTLPGM)
```

6. Create a store information record for the retail controller. Use option 2 (Work with retail store information) from the Configure Retail Communications Application menu to add new retail store information records.
7. Check to see if the line, controller, and devices are varied on by using the Work with Configuration Status command.

```
WRKCFGSTS CFGTYPE( LIN) CFGD(RTLLINE)
```

If the objects are not varied on, you can vary them on by selecting option 1 from the WRKCFGSTS display.

Switched Line Retail Controllers Configuration

This is a sample configuration for a 4680 retail controller on a switched (dial-up) line to the AS/400 system. The IBM 5853 modem may be used to perform single-line automatic dialing to compatible modems at 2400 bps. The dial is attempted when the ADCS emulation program attempts to establish a session with the retail controller. You may need to specify additional parameters to those shown for your communications environment.

1. Determine the AS/400 communications line resource name by using the Work with Hardware Resources command.

```
WRKHDWRSC
```

2. Configure a switched SNA primary communications line using the Create Line Description (SDLC) command.

```
CRTLNSDLC LIND(SWITCHLINE) RSRNAME(LIN 22) ROLE( PRI)
CNN( SWTPP) AUTOCALL( NO)
EXCHID( 56 1) SWTCNN( DIAL) AUTOANS( NO)
AUTODIAL( YES) DIALCMD( V25BIS) MAXFRAME(265)
```

3. Configure a retail communications controller using the Create Controller Description (Retail) command.

```
CRTCTLRTRL CTLD(POSC2) TYPE(468 ) MODEL( ) LINKTYPE( SDLC)
SWITCHED( YES) SWTLINLST(SWITCHLINE)
MAXFRAME(265) EXCHID( 4D 1) INLCNN( DIAL)
CNNNR('1,919,7623248') STNADR(C2)
```

The station address and exchange ID must match those specified when configuring the 4680 store system host communications.

4. Configure a retail device for HCP communications.

```
CRTDEVRTL DEVD(POSC2HCP) LOCADR( 1) RMTLOCNAME(POSC2HCP)
CTL(POSC2) MAXLENRU(256) APPTYPE( OTHER)
```

It is strongly recommended that the device name and remote location name be the same. This naming convention makes working with communications management much easier on the Work with Configuration Status system displays.

5. Configure a retail device for start user program using the Create Device Description (Retail) command.

```
CRTDEVRTL DEVD(POSC2SUP) LOCADR( 3) RMTLOCNAME(POSC2SUP)
CTL(POSC2) MAXLENRU(256) APPTYPE( SBMRTLPGM)
```

6. Create a store information record for the retail controller. Use option 2 (Work with retail store information) from the Configure Retail Communications Application menu to add new retail store information records.
7. Check to see if the line, controller, and devices are varied on by using the Work with Configuration Status command.

```
WRKCFGSTS CFGTYPE( LIN) CFGD(SWITCHLINE)
```

X.25 Retail Network Configuration

This is a sample configuration for a 4684 retail controller attached to an AS/400 system on an X.25 network. Use the commands discussed in "Nonswitched Retail Controllers Configuration" on page B-1 and "Switched Line Retail Controllers Configuration," as well as the following example, to configure an X.25 retail network.

```
CRTLINX25 LIND(RTLX25) RSRNAME(LIN 71) LGLGCHLE(( 1 PVC))
NETADR(4 76 38) CNNINIT( LOCAL) DFTPKTSIZE(256)
DFTWDWSIZE(7) TEXT(X.25 line description)

CRTCTLRTRL CTLD(C4684X25) TYPE(4684) MODEL( ) LINKTYPE( X25)
ONLINE( NO) LINE(RTLX25) EXCHID( 5 1) NETLVL(1984)
LGLCHLID( 1)

CRTDEVRTL DEVD(HCPDEV) LOCADR( 1) RMTLOCNAME(HCP) ONLINE( NO)
CTL(C4684X25)

CRTDEVRTL DEVD(RTLDEV) LOCADR( 2) RMTLOCNAME(RTL 2) ONLINE( NO)
CTL(C4684X25)
```

TCP/IP Configuration for Communication to 4690 RCO

This is a sample configuration for a 4690 retail controller attached to an AS/400 over Token Ring using TCP/IP protocol.

This configuration is for a AS/400 and a 4690 that are in the same subnet. This means No router configuration is needed. This example also expects a TRNLINE Token Ring line has been created.

1. Add the TCP interface using the ADDTCPIFC command.

```
ADDTCPIFC INTNETADR('1 .1.1 .1 ') LIND(TRNLINE)
SUBNETMASK('255.255.255. ')
```

2. Add the TCP Host table entry using the ADDTCPHTE command.

Note: If the Domain Name server is configured, this step may not be necessary. If you do specify a host table entry for your 4690 Retail Connectivity Option server, you may want to change the Searched first parameter of the Remote Name Server configuration to *LOCAL. The Remote Name Server configuration can be found specifying option 13 off the CFGTCP menu or the TCPCFG command.

```
ADDTCPHTE INTNETADR('1 .1.1 .12') HOSTNAME((ADXLCCN))
```

3. Update 4690 TCP configuration file (adxipccz.bat). For more information see the 4690 TCP configuration manual.

Note: The route configuration is not needed to connect to the AS/400.

```
adxhsi21
ifconfig lan 1 .1.1 .12 netmask 255.255.255.
route add default 1 .1.1 .1 1
route add default 1 .1.1 .2 2
```

4. Start the RCO server on the 4690. In this example the default RCO port is being started(20000). Just starting the RCO server without the -P flag specified would have the same effect.

```
adxsrvt1 -P2
```


Appendix C. Utility Trailer Record

The Retail Application Interface Suite sometimes places a utility trailer record at the end of a file. This record is used by some of the functions in the utility, and it contains data that you may want to use in building your applications. It is ignored by the CVTFRMRTL command and not placed into your translated file.

The functions of the utility that create the record are:

- RCVRTLF** Receive Retail File Command
- CVTTORTLF** Convert to Retail File Command (when FMTOPT(*SNDRTL) is specified)
- HCP-LOAD** The load function of HCP emulation

The functions of the utility that use the record are:

- SNDRTL** Send Retail File Command
- HCP-DUMP** The dump function of HCP emulation

If the record is not present for SNDRTL and HCP-DUMP, they assume that the last sector of data is a complete 256-byte sector.

If created, the record is the last record in the file. If the RCVRTL command is used multiple times with the MBROPT(*ADD) option to the same output member, multiple trailers will be scattered throughout the data. If the CVTTORTLF command is used multiple times to the same output member, only one trailer record will exist at the end of

the file. The byte offset will reflect the combined effects of each operation taking blocking and spanning into account. The format of the trailer record follows:

Byte(s)	Contents
1	X'FF'
2-8	Character data '5728CF1'
9	X'FF'
10-12	Byte offset of last significant byte in last sector. 0 indicates that all 256 bytes are significant. This is a zoned decimal number.
14-18	Number of sectors transmitted. Not loaded by CVTTORTLF. This is a zoned decimal number.
20-24	Last sector number transmitted. Not loaded by CVTTORTLF. This is a zoned decimal number.
26-31	HCP file name. This is the 6-character name used for transmission. Not loaded by CVTTORTLF.

The number of sectors transmitted and last sector number transmitted are useful to you if you intend to do a **trickle transmit**. A trickle transmit means to send data through communications lines little by little. The last sector number transmitted should be returned to HCP on subsequent RCVRTL commands as the STRDTABLK.

Appendix D. QRTXHASH Randomizing Routine

These instructions are needed to use the QRTXHASH randomizing routine on the Retail Application Interface Suite.

1. Rename the direct file on the AS/400 system to QARDIREC and make sure it is in the QRAIS library. The direct file must be a retail file that has previously been received from the 4680 retail controller.

Note: The Override with Database File (OVRDBF) command can be used instead of renaming the direct file.

2. CALL PGM(QRAIS/QRTHSHCR). This creates all the other files needed by the main program, QRTXHASH. This is a CL program which creates the files if they are not there, but deletes and creates the files again if they already exist.

The files created are QARTKEYF, QARTFREE, QARTOVER, QARTWORK, QARTNULLWK, QART4680, QARTMERG, and QARTKEYED.

Note: If the AS/400 system is putting security level 30 or 40 into effect, users calling the QRTHSHCR program must have explicit authority to library QRAIS (meaning that they have *ALL authority to the library). This is required since the QRTHSHCR program deletes and creates files in the QRAIS product library. This program only needs to be run once to create work files. After the work files have been created, program QRTXHASH can be called to convert direct files to keyed files as often as needed without running program QRTHSHCR again.

If you have Version 2 Release 3 or later releases installed, the Retail Application Interface Suite for AS/400 will work with a security level at 50.

3. CALL PGM(QRAIS/QRTXHASH) PARM(X'000' X'00000' X'XXXXXXXXXXXXXXXXXXXXX' X'000' X'00000' X'0')

For example,

```
CALL PGM(QRAIS/QRTXHASH) PARM(' 46' ' 1443' ' QARTKEYED
' 6' ' 4' ' ')
```

- a. The first parameter is the logical record length of the keyed records (range is 4-255 and length is 3 positions).

- b. The second parameter is the number of logical records to process (range is 1 to the total number of keyed records in the direct file and length is 5 positions).
- c. The third parameter is the full name of the keyed file (length is 18 characters; all 18 characters must be entered, even if they are blanks).
- d. The fourth parameter is key length (range 4-255 and length is 3 positions).
- e. The fifth parameter is chaining threshold (range is 1-99999 and length is 5 positions).
- f. The sixth parameter must always be zero (length is 1 position). This parameter initializes the internal call parameters.

Note: Each parameter position must be filled to use QRTXHASH, even if they are zeros or blanks.

4. After QRTXHASH has completed, rename file QARTKEYED located in QRAIS. The next time QRTXHASH is run, it will erase all the files, excluding QARDIREC. The renamed file is ready to be sent to the 4680 retail controller to be used as a keyed file.

Return Codes

Return Code	Description
1A	First parameter not numeric
1B	First parameter not in range 4-255
2A	Second parameter not numeric
2B	Second parameter equal to zero
4A	Fourth parameter not numeric
4B	Fourth parameter not in range 4-255
5A	Fifth parameter not numeric
5B	Second parameter equal to zero
80	Program ended abnormally. Make a copy of file QARDIREC and the call parameters. Contact your IBM representative.

For additional return codes, see the *Retail Communications Programming* book, SC41-3448, or the *COBOL/400 Reference*, SC09-1813.

Appendix E. AS/400 System Error Codes

The following appendix has a list of possible error codes that may be returned by the RAIS API. For UNIX error codes you may also refer to one of the following: *AS/400 Integrated File System Introduction* or *AS/400 System API Reference*.

AS/400 UNIX Error Codes

EDOM	3001 : Domain error in math function.	ENUMMBRS	3019 : No ftell if more than 1 member.
ERANGE	3002 : Range error in math function.	ENUMRECS	3020 : No ftell if too many records.
ETRUNC	3003 : Truncation on I/O operation.	EINVAL	3021 : A specified parameter was not valid. Check that the positional parameters are specified in the correct order and that the corresponding Flags are specified.
ENOTOPEN	3004 : The file descriptor is referencing a closed file.	EBADFUNC	3022 : Invalid function pointer.
ENOTREAD	3005 : The file descriptor is referencing a file that has not been opened for read.	ENOENT	3025 : The source 4690 file does not exist. For HCP, it is possible that the name specified could not be converted to an HCP logical name.
EIO	3006 : A communications IO error occurred. Generally more data was returned than was expected, or the link went down.	ENOREC	3026 : The specified record key was not found in the retail application file.
ENODEV	3007 : The location specified was not correct, or a route to the location does not exist.	EPERM	3027 : Operation not permitted. This may also occur if you have a licensed version of the Suite and your license is no longer valid. Please contact the provider to get a new license key.
ERECIO	3008 : File opened for record I/O.	EBADDATA	3028 : Message data invalid.
ENOTWRITE	3009 : The file descriptor is referencing a file that has not been opened for write.	EBUSY	3029 : The 4690 resource is busy. Some other application is using the file.
ESTDIN	3010 : stdin cannot be opened.	EBADOPT	3040 : Bad option on I/O function.
ESTDOUT	3011 : stdout cannot be opened.	ENOTUPD	3041 : File not opened for update.
ESTDERR	3012 : stderr cannot be opened.	ENOTDLT	3042 : File not opened for delete.
EBADSEEK	3013 : The offset is referencing a displacement in the file that is not valid.	EPAD	3043 : Padding occurred on write operation.
EBADNAME	3014 : Invalid file name specified.	EBADKEYLN	3044 : The key length or record length specified is not correct.
EBADMODE	3015 : Invalid file mode specified.	EPUTANDGET	3080 : Illegal write after read.
EBADPOS	3017 : Invalid position specifier.	EGETANDPUT	3081 : Illegal read after write.
ENOPOS	3018 : No record at specified position.	EIOERROR	3101 : I/O exception non-recoverable error.
		EIORECERR	3102 : I/O exception recoverable error.
		EACCES	3401 : The 4690 server can not access the file.

ENOTDIR	3403 : Not a directory.	EISCONN	3431 : A connection has already been established.
ENOSPC	3404 : No space available.	EMSGSIZE	3432 : Message size out of range.
EXDEV	3405 : Improper link.	ENETDOWN	3433 : The network is not currently available.
EWOULDBLOCK	3406 : Operation would have caused the process to be suspended.	ENETRESET	3434 : A socket is connected to a host that is no longer available.
EAGAIN	3406 : Operation would have caused the process to be suspended.	ENETUNREACH	3435 : A connection to the remote location could not be established.
EINTR	3407 : Interrupted function call.	ENOBUFS	3436 : There is not enough buffer space for the requested operation.
EFAULT	3408 : The address used for an argument was not correct.	ENOPROTOPT	3437 : The protocol does not support the specified option.
ENXIO	3415 : No such device or address.	ENOTCONN	3438 : Requested operation requires a connection.
EADDRINUSE	3420 : Address already in use.	ENOTSOCK	3439 : The specified descriptor does not reference a socket.
EADDRNOTAVAIL	3421 : Address not available.	EOPNOTSUPP	3440 : Operation not supported.
EAFNOSUPPORT	3422 : The type of socket is not supported in this protocol family.	EPFNOSUPPORT	3441 : The socket protocol family is not supported.
EALREADY	3423 : Operation already in progress.	EPROTONOSUPPORT	3442 : No protocol of the specified type and domain exists.
ECONNABORTED	3424 : A communication error occurred that caused the connection to drop.	EPROTOTYPE	3443 : The socket type or protocols are not correct.
ECONNREFUSED	3425 : A connection to the remote location was refused by the remote location. Check the status of the server on the remote system. For LU6.2(*SNA) check the transaction program name is correct.	ERCVDERR	3444 : The remote application sent an application error notification. Check the 4690 OS messages to see why the remote application could not access files on its local area network. This error occurs when there is no file server running in a Multi-Controller 4690 system.
ECONNRESET	3426 : A connection with a remote socket was reset by that socket.	ESHUTDOWN	3445 : Cannot send data after a shutdown.
EDESTADDRREQ	3427 : Operation requires destination address.	ESOCKTNOSUPPORT	3446 : The specified socket type is not supported.
EHOSTDOWN	3428 : A remote host is not available.	ETIMEDOUT	3447 : The remote location did not respond within the timeout period. Check the device status. Check the remote server. The server could be down or in use by another application.
EHOSTUNREACH	3429 : A route to the remote host is not available. For LU6.2(*SNA), a non IBM supplied mode must be configured using the Pre-established sessions parameter. For LU0(*HCP) and LU6.2(*SNA) make sure the configuration objects are in the proper state.		
EINPROGRESS	3430 : Operation in progress.		

EUNATCH	3448 : The protocol required to support the specified address family is not available at this time.	EITERBAD	3475 : Iterator is invalid.
EBADF	3450 : The file descriptor used is not valid. If the descriptor was valid previously, this may have been invalidated by the RCO server because of file access timeout. See the RCO server file table support to change the RCO file access timer.	EDAMAGE	3484 : A damaged object was encountered.
EMFILE	3452 : Too many open files for this process.	ELOOP	3485 : A loop exists in the symbolic links.
ENFILE	3453 : Too many open files in the system.	ENAMETOOLONG	3486 : A path name is too long.
EPIPE	3455 : Broken pipe.	ENOLCK	3487 : No locks available.
EEXIST	3457 : The 4690 destination file already exists.	ENOTEMPTY	3488 : Directory not empty.
EDEADLK	3459 : Resource deadlock avoided.	ENOSYSRSC	3489 : System resources not available to complete request.
ENOMEM	3460 : No memory is available on the 4690 or no memory available on the AS/400.	ECONVERT	3490 : Conversion error.
EOWNERTERM	3462 : The mutex no longer exists because the owner is no longer running.	E2BIG	3491 : Argument list too long.
EDESTROYED	3463 : The mutex was destroyed.	EILSEQ	3492 : Conversion stopped due to input character error.
ETERM	3464 : A Communications return code was returned that was not expected, so the connection was dropped.	ENOTENROLL	3498 : User not enrolled in system distribution directory.
EMLINK	3468 : Maximum link count for a file was exceeded.	EOFFLINE	3499 : Object is suspended.
ESPIPE	3469 : Seek request not supported for object.	EROOBJ	3500 : Object is a read only object.
ENOSYS	3470 : Function not implemented.	ELOCKED	3506 : The 4690 server can not access the file.
EISDIR	3471 : Specified target is a directory.	EFBIG	3507 : Object too large.
EROFS	3472 : Read-only file system.	EIDRM	3509 : The semaphore, shared memory, or message has been removed.
EUNKNOWN	3474 : Unknown system state.	ENOMSG	3510 : The queue does not contain a message of the desired type.
		EFILECVT	3511 : File ID conversion of a directory failed.
		EBADFID	3512 : The Keyed file specified is not in the correct state for processing. This will occur when you are trying to open a sequential file as a keyed file.
		ESTALE	3513 : File or object handle rejected by server.
		ESRCH	3515 : No such process.
		ENOTSIGINIT	3516 : Process not enabled for signals.
		ECHILD	3517 : No child process.

Appendix F. General Sales Application Cross Reference

This appendix maps the **General Sales Application** to the QUSRPOSGS Collection's predefined table definitions installed with RAIS option 10. This mapping is also shown by the QZCFGSDS template defined that is included as part of the option 10 installation.

The mapping is as follows:

<i>Table F-1. String Description.</i>			
Source Information		Target Information	
Field	Bit Number	Column Name	Column Type
FFFFFFFF	BBBBBB	TTTTTTT	CCCCCC

Where:

Field	The name of the field from which the row is derived.
Bit Number	The number of the bit from which the row is derived.
Table	The name of the table which now contains the data.
Column	The name of the column in the table which now contains the data.

Notes:

1. The definition table may not refer to all of the fields in every case.
2. There may not be mapping in all cases.
3. This reference information can be found in the programming guide for each application.

EALTRANS - GSA Transaction Summary Log

This appendix is a cross reference between detail files defined by the 4680 General Sales Application and associated features and the QUSRPOSGS relational database tables.

Reference:

- SC30-3631 4680-4690 General Sales Application: Programming Guide
- SC30-3514 4680 General Sales Application: Electronic Funds Transfer (EFT)
- SC30-3461 4680 Price Management Feature: User's Guide

String '00' Transaction Log Header to QAZCFGHDR

<i>Table F-2 (Page 1 of 2). String '00' Transaction Log Header.</i>			
Source Information		QAZCFGHDR Information	
Field	Bit Number	Column Name	Column Type
system defined		STORE#GEN	char(6) (Key)
STRINGTY		discarded	
TERMINAL		TERMINAL	char(4) (Key)
OPERATOR		OPERATOR	char(10)
TRANSNUM		TRANSNUM	integer (Key)
	Set to IN0OFFLINE value (0,1)	KEYEXTEND	short integer (Key)
TRANTYPE		TRANTYPE	short integer
NUMSTRIN		NUMSTRIN	short integer
GROSSPOS		GROSSPOS	decimal(15,0)
GROSSNEG		GROSSNEG	decimal(15,0)
NETCASH	negated by IN0NNCASH flag	NETCASH.	decimal(15,0)
NETNCASH	negated by IN0NNNCASH flag	NETNONCASH	decimal(15,0)
PASSWORD	To discard, set the template Retail data type to drop(6).	PASSWORD	integer
DATE		TDATE	date (Key)
TIME		TTIME	time (Key)
INDICAT0.HDR	Bit15 X'8000' Must be Zero	INDICAT0	char(1)
INDICAT0.HDR	Bit14 X'4000' Net cash field negative	IN0NNCASH	numeric(1,0)
INDICAT0.HDR	Bit13 X'2000' Net noncash field negative	IN0NNNCASH	numeric(1,0)
INDICAT0.HDR	Bit13 X'2000' Net noncash field negative	IN0NNNCASH	numeric(1,0)
INDICAT0.HDR	Bit12 X'1000' Store initiated layaway cancel	IN0LYSTORC	numeric(1,0)
INDICAT0.HDR	Bit11 X'0800' Operator totals kept in the accounting file	IN0OPRTTL	numeric(1,0)
INDICAT0.HDR	Bit10 X'0400' Manager's key used during transaction	IN0MGRKEY	numeric(1,0)
INDICAT0.HDR	Bit9 X'0200' Transaction number set prior to transaction	IN0SETTRN	numeric(1,0)
INDICAT0.HDR	Bit8 X'0100' Send transaction	IN0TSEND	numeric(1,0)
INDICAT0.HDR	Bit7 X'0080' Offline transaction header	IN0OFFLINE	numeric(1,0)
INDICAT0.HDR	Bit6 X'0040' Reentry of offline transaction	IN0REENTRY	numeric(1,0)
INDICAT0.HDR	Bit5 X'0020' Transaction suspended	IN0SSPND	numeric(1,0)
INDICAT0.HDR	Bit4 X'0010' Transaction retrieved from suspension	IN0RTRVD	numeric(1,0)
INDICAT0.HDR	Bit3 X'0008' Transaction voided	IN0VOIDED	numeric(1,0)
INDICAT0.HDR	Bit2 X'0004' Terminal initialized before or during transaction	IN0TRMINIT	numeric(1,0)
INDICAT0.HDR	Bit1 X'0002' Terminal enabled prior to transaction	IN0TRMON	numeric(1,0)
INDICAT0.HDR	Bit0 X'0001' New operator from previous transaction	IN0NEWOP	numeric(1,0)
INDICAT1.HDR	Bit15 X'8000'	Discarded	
INDICAT1.HDR	Bit14 X'4000'	Discarded	
INDICAT1.HDR	Bit13 X'2000'	Discarded	

General Sales Cross Reference

<i>Table F-2 (Page 2 of 2). String '00' Transaction Log Header.</i>			
Source Information		QAZCFGHDR Information	
INDICAT1.HDR	Bit12 X'1000'	Discarded	
INDICAT1.HDR	Bit11 X'0800'	Discarded	
INDICAT1.HDR	Bit10 X'0400'	Discarded	
INDICAT1.HDR	Bit9 X'0200'	Discarded	
INDICAT1.HDR	Bit8 X'0100'	Discarded	
INDICAT1.HDR	Bit7 X'0080' Reserved	INDICAT1	char(1)
INDICAT1.HDR	Bit6 X'0040' Void previous by line item	IN1VPLI	numeric(1,0) Note: The TranType does NOT reflect what is in the programmers guide. The programmers guide implies a void previous transaction by line item has a TRANTYPE of 61. A CASH transaction (TRANTYPE 1) can be created as a void previous transaction by line item.
INDICAT1.HDR	Bit5 X'0020' Customer initiated layaway cancel	IN1LYCUSTC	numeric(1,0)
INDICAT1.HDR	Bit4 X'0010' Void layaway cancel	IN1VOIDLYC	numeric(1,0)
INDICAT1.HDR	Bit3 X'0008' Customer initiated layaway cancel line items	IN1LYVPLI	numeric(1,0)
INDICAT1.HDR	Bit2 X'0004' Vat tax transaction	IN1VATTAX	numeric(1,0)
INDICAT1.HDR	Bit1 X'0002' Layaway training transaction	IN1LYTRAIN	numeric(1,0)
INDICAT1.HDR	Bit0 X'0001' Suspended transaction has a string type 15 following this header	IN1SSPND15	numeric(1,0)
RINGTIME		discarded	
TENDTIME		discarded	
NOSLTIME		discarded	
INACTIME		discarded	
NETCASH.CURR		decimal(15,0)	
<p>Note: The Primary Keyed field order in the QAZCFGHDR file is:</p> <p>STORE##GEN TERMINAL TRANSNUM TDATE TTIME KEYEXTEND</p>			

String '00' Transaction Log Header to QAZCFGTTIM

Source Information		QAZCFGTTIM Information	
Field	Bit Number	Column Name	Column Type
system defined		STORE#GEN	char(6)
STRINGTY		discarded	
TERMINAL		TERMINAL	char(4)
OPERATOR		OPERATOR	char(10)
TRANSNUM		TRANSNUM	integer
	1	KEYEXTEND	short integer
TRANTYPE		discarded	
NUMSTRIN		discarded	
GROSSPOS		discarded	
GROSSNEG		discarded	
NETCASH		discarded	
NETNCASH		discarded	
PASSWORD		discarded	
DATE		TDATE	date
TIME		TTIME	time
INDICAT0.HDR		discarded	
INDICAT1.HDR		Discarded	
RINGTIME		RINGTIME	integer
TENDTIME		TENDTIME	integer
NOSLTIME		NOSLTIME	integer
INACTIME		INACTIME	integer
NETCASH.CURR		discarded	
Note: The Primary Keyed field order in the QAZCFGTTIM file is: NONE			

General Sales Cross Reference

String '01' Header String Extension to QAZCFGHDRX

This string is included for re-entry, return, void previous by line (VPLI) item, or transactions retrieved from suspension.

String 01 for Re-Entry

Source Information		QAZCFGHDRX Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
STRINGTY		discarded	
TERMINAL		ORGTERMINAL	CHAR(4)
OPERATOR		ORGOPERATR	char(10)
TRANSNUM		ORGTRNSNM	integer
DATE		ORGTDATE	date
TIME		ORGTTIME	time

String '02' Price Change to QAZCFGPRCH

This string is included after price change transactions only.

Source Information		QAZCFGPRCH Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
STRINGTY		discarded	
ITEMCODE		ITEMCODE	char(26)
NEWSQUAN		NEWSQUAN	decimal(10,2)
NEWPRICE		NEWPRICE	decimal(15,0)
OLDSQUAN		OLDSQUAN	decimal(10,2)
OLDMETHO		OLDMETHO	numeric(1,0)
OLDPRICE		OLDPRICE	decimal(15,0)

String '03' Totals String to QAZCFGTOTL

This string is included for totals reset transactions. There are three types of readout requests possible.

- 1 = Totals reset and close
- 2 = Totals reset
- 9 = Cancel reset

GSA Design note

Option 9 yields a header string only.
Option 1 or 2 creates a string 3, but there are no INDICAT bits to tell them apart.

GSA Version Two

NETCASHC.TSS is the accumulated domestic value of all foreign currencies in the cash drawer. Unlike currencies are accumulated into the same field therefore it is meaningless and discarded.

AMTCCASH.CUR is the accumulated counted face value of foreign currencies.

Table F-6. String '03' Totals String.

Source Information		QAZCFGTOTL Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
STRINGTY		discarded	
GROSSPOS		GROSSPOS	decimal(15,0)
GROSSNEG		GROSSNEG	decimal(15,0)
NETCASH		NETCASH	decimal(15,0)
NETCASHC.TSS	Accumulation of multiple foreign currencies.	NETCASHC	decimal(15,0)
NETNCASH		NETNCASH	decimal(15,0)
AMTLOAN		AMTLOAN	decimal(15,0)
AMTWITHD		AMTWITHD	decimal(15,0)
AMTCCASH		AMTCCASH	decimal(15,0)
AMTCCASH.CUR		AMTCCASH	decimal(15,0)
INDICAT0	Bit7 X'0080'	indicat0	char(1)
INDICAT0	Bit6 X'0040' Net cash negative	in0cashn	numeric(1,0)
INDICAT0	Bit5 X'0020' Net noncash negative	in0ncashn	numeric(1,0)
INDICAT0	Bit4 X'0010' Loan amount negative	in0loann	numeric(1,0)
INDICAT0	Bit3 X'0008' Withdrawal amount negative	in0withdn	numeric(1,0)
INDICAT0	other	in0rsrv	char(3)

General Sales Cross Reference

String '04' Cash Drawer Change String to QAZCFGCS DW

This string is included for loan, withdrawal, and cash count transactions.

GSA Version Two

For a cash count transaction, the AMTTEN01 field contains the face value of the currency and field AMTCURR01 contains the domestic value for this entry. All reconciliation is performed using domestic values, therefore the face value of the foreign tender is NOT inserted into the table.

```
If AMTCURR01 doesn't exist or is equal to AMTTEND01
  THEN
    INSERT AMTTEND01 into TILLCHG.AMT
  ELSE
    INSERT AMTCURR01 into TILLCHG.AMT
```

Table F-7. String '04' Cash Drawer Change String.

Source Information		QAZCFGCS DW Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE##GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
STRINGTY		discarded	
TENDERxx		TENDER	short integer Table H-1 on page H-1
AMTTENxx		AMTTEND	decimal(15,0)
AMTCURRxx		AMTCURR	decimal(15,0)
Note: For string 04, there exists a tender entry for each tender. A record entry in QAZCFGCS DW will exist for each tender entry.			

String '05' Store Close to QAZCFGCS CLOS

This string is written without a header string when a DISABLE ALL executes.

Table F-8. String '05' Store close.

Source Information		QAZCFGCS CLOS Information	
Field	Bit Number	Column Name	Column Type
system defined		STORE##GEN	char(6) (Key)
DATE		TDATE	date (Key)
TIME		TTIME	time (Key)

String '06' Merchandise / Non Merchandise String to QAZCFGMRCH

This string is included for each merchandise/nonmerchandise line item entry for sales transactions only.

Source Information		QAZCFGMRCH Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY		discarded	
ITEMCODE		ITEMCODE	char(26)
DEPARTME		DEPARTME	char(6)
AMTITEM		AMTITEM	decimal(16,0)
TAXPLAN	Bit 7 X'80'	taxplan	char(1)
TAXPLAN	Bit 6 X'40' Vat Tax 7	taxtbl7	numeric(1,0)
TAXPLAN	Bit 5 X'20' Vat Tax 6	taxtbl6	numeric(1,0)
TAXPLAN	Bit 4 X'10' Vat Tax 5	taxtbl5	numeric(1,0)
TAXPLAN	Bit 3 X'08' Tax table 1/Vat Tax 1	taxtbl1	numeric(1,0) (1=Normal tax, 2=Vat tax)
TAXPLAN	Bit 2 X'04' Tax table 2/Vat Tax 2	taxtbl2	numeric(1,0) (1=Normal tax, 2=Vat tax)
TAXPLAN	Bit 1 X'02' Tax table 3/Vat Tax 3	taxtbl3	numeric(1,0) (1=Normal tax, 2=Vat tax)
TAXPLAN	Bit 0 X'01' Tax table 3/Vat Tax 4	taxtbl4	numeric(1,0) (1=Normal tax, 2=Vat tax)
INDICAT0.MES	Bit 15 X'8000'	Discarded	
INDICAT0.MES VPLI	Bit 14 X'4000'	in0vpli	numeric(1,0)
INDICAT0.MES	Bit 13 X'2000'	in0versn2	numeric(1,0)
INDICAT0.MES	Bit 12 X'1000' Additional data follows	in0dataent	numeric(1,0)
INDICAT0.MES	Bit 11 X'0800' Quantity key	in0quanent	numeric(1,0)
INDICAT0.MES	Bit 10 X'0400'	in0multipr	numeric(1,0)
INDICAT0.MES	Bit 9 X'0200'	in0offline	numeric(1,0)
INDICAT0.MES	Bit 8 X'0100'	in0nontax	numeric(1,0)

General Sales Cross Reference

Source Information		QAZCFGMRCH Information	
INDICAT0.MES	Bit 7 X'0080' Item Voided	in0voided	numeric(1,0) V2 Design note If the Sales Associate voids an ITEM from a VPLI transaction then the TITEM.Voided flag is set to default=0 as if the ITEM were sold.
INDICAT0.MES	Bit 6 X'0040' Item Returned	in0return	numeric(1,0)
INDICAT0.MES	Bit 5 X'0020' Extended amount negative	in0negamt	numeric(1,0)
INDICAT0.MES	Bit 4 X'0010' Keep item movement	in0itmvmmt	numeric(1,0)
INDICAT0.MES	Bit 3 X'0008' non-merchandise	in0nonmrch	numeric(1,0)
INDICAT0.MES	Bit 2 X'0004' keyed item code	in0itmkeyd	numeric(1,0)
INDICAT0.MES	Bit 1 X'0002' item data wanded	in0itmwnd	numeric(1,0)
INDICAT0.MES	Bit 0 X'0001' Item amount from file	in0amtfrmf	numeric(1,0) V2 Design note The PriceOver indicates if item price was not from file.
INDICAT1.MES	Bits 7,6,5,4,3	indicat1	char(5)
INDICAT1.MES	Bit 2 X'04' Modify ticket price override	in1mdfyovr	numeric(1,0)
INDICAT1.MES	Bit 1 X'02' Modify ticket original data string follows	in1mdfyorg	numeric(1,0)
INDICAT1.MES	Bit 0 X'01'	in1rsrv	char(1)

String '07' Merchandise / Non Merchandise String Extension to QAZCFGMRGX

This string is present when any of the bits 10 through 13 are set in INDICAT0.MES of the merchandise/non-merchandise string. It occurs after the merchandise string only.

V2 Design Note:

The RECID##GEN column is used to maintain the order of items within a transaction. GSA and SA applications ensure the Item quantity, Item price override, or Item weight extension string immediately follows the Merchandise string in the transaction log. Therefore the RECID##GEN column for the QAZCFGMRGX table is one greater than the RECID##GEN value for the associated QAZCFGMRCH record.

GSA Design Note

General Sales Application doesn't provide the Item Record information containing the pricing method within the TLOG string data.

Table F-10. String '07' Merchandise / Non Merchandise String Extension.

Source Information		QAZCFGMRGX Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE##GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
system defined		RECID##GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY		discarded	
GROUP		EXTGROUP	short integer
QUANTITY	Number of Items	QUANTITY	decimal(10,2)
<div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p style="text-align: center;">V2 Design Note</p> <p>The number of items is divided by 100 prior to insertion. e.g. Quantity of 500 = EntQty of 5.</p> </div>			
SALEQUAN		SALEQUAN	decimal(10,2)
SALEPRIC	This field is the concatenated Deal Price and Unit price and requires the Item Record pricing method to parse.	SALEPRIC	decimal(15,0)
QUANOV	Quantity Override	QUANTOV	decimal(10,2)
PRICEOV	Price Override	PRICEOV	decimal(15,0)
<div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p style="text-align: center;">V2 Design Note</p> <p>This field is not implemented in GSA. If it were, this price would represent the price keyed by the operator. This information is already captured in String 6 as the AMTITEM extended price.</p> </div>			

General Sales Cross Reference

String '08' Miscellaneous String to QAZCFGMISC

This string is present for all transactions that use tax codes, allowances, fees, deposits or payments.

String '08' Miscellaneous String (For Tax Codes): If INDICAT0.NME bit 5, 6 or 7 then CUSTOMER contains the TAXCODE.

Source Information		QAZCFGMISC Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY		discarded	
CUSTOMER	Taxcode	CUSTOMER	char(24)
AMOUNT		AMOUNT	decimal(15,0)
TAXCODE		TAXCODE	short integer
INDICAT0	Bits 15,14,13,12	indicat0	char(4)
INDICAT0	Bit 11 X'0800' Total Key Pressed	in0totalk	EvntdType=(6)
INDICAT0	Bit 10 X'0400' NonTax Key Pressed	in0nontaxk	
INDICAT0	Bit 9 X'0200' Void	in0void	Voided
INDICAT0	Bit 8 X'0100' Return	in0return	Returned
INDICAT0	Bit 7 X'0080' No tax	in0notax	NoTaxCode
INDICAT0	Bit 6 X'0040' Auto Tax	in0autotax	AutoTax
INDICAT0	Bit 5 X'0020' Manual Tax	in0mantax	ManualTax
INDICAT0	Bit 4 X'0010' Allowance	in0allownc	
INDICAT0	Bit 3 X'0008' Send Fee	in0sendfee	
INDICAT0	Bit 2 X'0004' Other Fee	in0other	
INDICAT0	Bit 1 X'0002' Payment	in0payment	
INDICAT0	Bit 0 X'0001' Deposit	in0deposit	

String '09' Data Entry to QAZCFGDTAE

This string is created any time magnetic ticket data is read with the sale of an item, or when the sales associate enters item miscellaneous data.

<i>Table F-12. String '09' Data Entry.</i>			
Source Information		QAZCFGDTAE Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY		discarded	
DATA		DATA	char(30)

General Sales Cross Reference

String '10' Discount to QAZCFGDISC

This string is present for line item or transaction discounts for sales transactions only. INDICAT0 Bit 3 indicates if this string is a Transaction or an Item Discount.

Source Information		QAZCFGDISC Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY		discarded	
ENTRYCOD	Reason	ENTRYCOD	char(6)
AMTDISCO		AMTDISCO	decimal(15,0)
PERDISCO	Rate	PERDISCO	decimal(5,2)
INDICAT0	Bits 7,6,5,4	indicat0	char(4)
INDICAT0	Bit 3 X'08' Transaction discount	in0trandis	numeric(1,0)
INDICAT0	Bit 2 X'04' NonTax Key Pressed	in0nontaxk	numeric(1,0)
INDICAT0	Bit 1 X'02' Void	in0void	numeric(1,0)
INDICAT0	Bit 0 X'01' Return	in0return	numeric(1,0)

String '11' Cash Tender String to QAZCFGTNRD

This string is present for sales transactions only.

Source Information		QAZCFGTNRD Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY		discarded	
0		TENDERTY	short integer
AMTTENDE		AMTTENDE	decimal(15,0)

String '12' Cash-Like Tender String to QAZCFGTNDR

This string is present for all cash-like tenders and whenever change is returned for sales transaction only.

Design Note

If INDICAT0.TEN bit 10 is set this is a change string.

GSA Version Two

The cash-like tender string has been extended to support foreign currency and user data. Change is always be given the customer in domestic currency. Therefore the AMTEND.CURR field is discarded when INDICAT0.TEN bit 10=1.

Table F-15 (Page 1 of 2). String '12'Cash-Like Tender String.

Source Information		QAZCFGTNDR Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE##GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
system defined		RECID##GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY		discarded	
TENDERTY		TENDERTY	short integer
AMTTENDE		AMTTENDE	decimal(15,0)
CUSTOMER		CUSTOMER	char(24)
STATUS		STATUS	short integer
INDICAT0	Bit 15 X'8000'	indicat0	char(1)
INDICAT0	Bit 14 X'4000' approved	in0approve	numeric(1,0)
INDICAT0	Bit 13 X'2000' disapproved	in0disaprv	numeric(1,0)
INDICAT0	Bit 12 X'1000' rejected and alternate payment method accepted	in0altpay	numeric(1,0)
INDICAT0	Bit 11 X'0800' Tender action suspended	in0suspend	numeric(1,0)
INDICAT0	Bit 10 X'0400'	in0negativ	numeric(1,0) This bit is used to indicate that the cash like tender string represents CHANGE to the customer.
INDICAT0	Bit 9 X'0200' Tender cash to charge	in0cshtcrg	numeric(1,0)
INDICAT0	Bit 8 X'0100' Preferred customer	in0prfcst	numeric(1,0)
INDICAT0	Bit 7 X'0080'	in0fillimit	numeric(1,0) FloorLimit (Set flag)
INDICAT0	Bit 6 X'0040'	in0instcre	numeric(1,0) In Store Credit (Set Flag)
INDICAT0	Bit 5 X'0020'	in0oprprv	numeric(1,0)

General Sales Cross Reference

<i>Table F-15 (Page 2 of 2). String '12'Cash-Like Tender String.</i>			
Source Information		QAZCFGTNRD Information	
INDICAT0	Bit 4 X'0010'	in0creddpt	numeric(1,0) Credit Dept (Set Flag)
INDICAT0	Bit 3 X'0008'	in0hstcred	numeric(1,0) Host Credit (Set Flag)
INDICAT0	Bit 2 X'0004'	in0oostndr	numeric(1,0) Out of Store Credit (Set Flag)
INDICAT0	Bit 1 X'0002'	in0oosbku	numeric(1,0) Out of Store Backup (Set Flag)
INDICAT0	Bit 0 X'0001'	in0offline	numeric(1,0) Offline (set Flag)
INDICAT1	Bits 8,7,6,5,4,3	indicat1	char(6)
INDICAT1	Bit 1 X'0002'	in1noprint	numeric(1,0) Don't print tender report
INDICAT1	Bit 0 X'0001'	in1userdta	numeric(1,0) Packed user data
AMTEND.CURR		AMTTENDCUR	amt. ForeignAmt Note: This field is discarded when the string represents change.
USER.TDR		USRDATA	char(26) UserData

String '13' Tender Extension String to QAZCFGTNDX

This string is present for sales transactions only.

<i>Table F-16. String '13'Tender Extension String.</i>			
Source Information		QAZCFGTNDX Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE##GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
system defined		RECID##GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY		discarded	
CDAUTHOR	(Range 00-99)	CDAUTHOR	char(4)
TERMS		TERMS	char(15)

String '14' Tax String to QAZCFG TAX

This string is for sales transactions only, with one or more strings per transaction. Multiple strings exist when items are added to a transaction following the pressing of the Total key.

Source Information		QAZCFG TAX Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY		discarded	
1	AMTTAXA > 0	TAXTYPE	short integer
2	AMTTAXB > 0	TAXTYPE	short integer
3	AMTTAXC > 0	TAXTYPE	short integer
4	AMTTAXD > 0	TAXTYPE	short integer
AMTTAXA	TAXTYPE=(1); negated by in0aneg flag	AMT	decimal(15,0)
AMTTAXB	TaxpType=(2); negated by in0bneg flag	AMT	decimal(15,0)
AMTTAXC	TaxpType=(3); negated by in0cneg flag	AMT	decimal(15,0)
AMTTAXD	TaxpType=(4); negated by in0dneg flag	AMT	decimal(15,0)
INDICAT0	Bits 7,6,5,4	indicat0	char(4)
INDICAT0	Bit 3 X'08'	in0aneg	numeric(1,0) Refund - AMT set to negative
INDICAT0	Bit 2 X'04'	in0bneg	numeric(1,0) Refund - AMT set to negative
INDICAT0	Bit 1 X'02'	in0cneg	numeric(1,0) Refund - AMT set to negative
INDICAT0	Bit 0 X'01'	in0dneg	numeric(1,0) Refund - AMT set to negative
Note:			
Tax refund string processing sets the AMT to a negative number QAZCFG TAX table.			
A separate entry will be placed in the QAZCFG TAX table for each valid tax type.			

String '15' User Data String to QAZCFG USER

NOT PROCESSED

String '15' User data

Source Information		QAZCFG USER Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE#GEN	char(6) (Key)

General Sales Cross Reference

<i>Table F-18 (Page 2 of 2). String '15' User data.</i>		
Source Information	QAZCFGUSER Information	
Foreign Key Reference QAZCFGHDR	TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR	TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR	TDATE	date (Key)
Foreign Key Reference QAZCFGHDR	TTIME	time (Key)
Foreign Key Reference QAZCFGHDR	KEYEXTEND	short integer (Key)
system defined	RECID##GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY	discarded	
USER	USRDATA	char(20)

String '16' Log Time Analysis String to QAZCFGTTIM

This string is present whenever an operator signs on or off and when the inactive time reaches 10 minutes. The operator ID is 0 when an operator is not signed on. The maximum length of the log time analysis string is 28 bytes, plus up to 18 delimiters.

<i>Table F-19. String '16' Log Time Analysis String.</i>			
Source Information		QAZCFGTTIM Information	
Field	Bit Number	Column Name	Column Type
system defined		STORE##GEN	char(6)
STRINGTY		discarded	
TERMINAL		TERMINAL	char(4)
OPERATOR		OPERATOR	char(10)
0		TRANSNUM	integer
0		KEYEXTEND	short integer
RING.TIME		RINGTIME	integer
TEND.TIME		TENDTIME	integer
NONS.TIME		NOSLTIME	integer
INACT.TIME		INACTIME	integer
TIME		TTIME	time
DATE		TDATE	date
<p>Note: The Primary Keyed field order in the QAZCFGTTIM file is: NONE There is no header string with this string.</p>			

String '17' Void Previous Transaction String to QAZCFGVOID

Source Information		QAZCFGVOID Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
STRINGTY		discarded	
TERMINAL		ORGTERMINL	char(4)
AMOUNT		AMOUNT	decimal(15,0)
TRANSNUM		ORGTRANSNM	integer

String '18' Data Maintenance to QAZCFGDTAM

Source Information		QAZCFGDTAM Information	
Field	Bit Number	Column Name	Column Type
system defined		STORE#GEN	char(6) (Key)
STRINGTY		discarded	
OPERATOR		OPERATOR	char(10) (Key)
DATE		TDATE	date (Key)
TIME		TTIME	time (Key)
FUNCTION		FUNCTION	short integer
FILEID		FILEID	short integer
RESERVED		discarded	
DATA		DATA	char(100)
<p>Note: The Primary Keyed field order in the QAZCFGHDR file is:</p> <p>STORE#GEN OPERATOR TDATE TTIME</p> <p>There is no header string with this string.</p>			

General Sales Cross Reference

String '19' Invalid Sign-on String to QAZCFGSIGN

Successful sign-on attempts are combined with the normal transaction type selected as indicator bits in the header string. This string is for Invalid Sign-on attempts only. The reason code is captured into the TEVENT table. A HEADER entry is made for this string with a function type of '0015' (Operator Sign-on Attempted).

Source Information		QAZCFGSIGN Information	
Field	Bit Number	Column Name	Column Type
system defined		STORE##GEN	char(6) (Key)
STRINGTY		discarded	
OPERATOR		OPERATOR	char(10) (Key)
DATE		TDATE	date (Key)
TIME		TTIME	time (Key)
OPERATOR		OPERATOR	char(10)
PASSWORD		PASSWORD	integer
TERMINAL		TERMINAL	char(4)
TRANSNUM		TRANSNUM	integer
REASON	Reason code 77	REASON	decimal(2,0)
RESERVED		Discarded	

String '20' Modify Ticket to QAZCFGMDFY

This string is present if bit 1 or bit 2 of INDICAT1.MES in String 6 is on, or if the tax status is changed during a modify ticket keying sequence.

Source Information		QAZCFGMDFY Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE##GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
system defined		RECID##GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY		discarded	
TAXCOD		TAXCODE	decimal(15,0)
AMOUNT		ITEMMODIFY	decimal(15,0)

String '23' Multiple Currency to QAZCFGMCF

This string is present if bit 1 or bit 2 of INDICAT1.MES in String 6 is on, or if the tax status is changed during a modify ticket keying sequence.

Source Information		QAZCFGMCF Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
Foreign Key Reference QAZCFGHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY		discarded	
SUBSTRING	must = 99	discarded	
EXCHGRATE		EXCHGRATE	integer
ISOCURNUM		ISOCURNUM	integer
AMTTENDBAS		AMTTENDBAS	decimal(15,0)

General Sales Cross Reference

String '25' Full Screen Customer information to QAZCFGRCP

This string captures customer data necessary to create shipping labels for various types of transactions.

GSA Design Note

The GSA design concatenates the customer first and last name into the same field.

Source Information		QAZCFGRCP Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
STRINGTY		discarded	
NAME		NAME	char(34)
ADDRESS		ADDRESS	char(34)
CITY		CITY	char(20)
STATE/PROVINCE		STATE	char(2)
ZIPCODE		ZIP	char(5)
PHONE,NUMBER		PHONE	char(12)
ADDITIONAO,INF		ADDITIONAL	char(56)

String '26' Full Screen Send Delivery to QAZCFGSNL

String 26 provides information relative to the table used to calculate delivery charges. For example, UPS, Federal Express and AirBorne are examples of carriers who would have their own independent method of charging for services.

The current GSA function doesn't associate a string 26 with a string 25 (customer).

Table F-26. String '26' Full Screen Send Delivery.

Source Information		QAZCFGSNL Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE##GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
STRINGTY		discarded	
TABLE		ZNRTTBL	char(2)
WEIGHT		WEIGHT	CHAR(4)

General Sales Cross Reference

String '27' Tender Removal to QAZCFGTNDR

String 27 captures information that a tender removal was performed at a terminal. The HEADER string captures the transaction type (1008). A row of information is inserted into the QAZCFGTNDR table with the tender type information, all other columns of the QAZCFGTNDR table are set to their default values. e.g. AMTTENDE = 0.

Source Information		QAZCFGTNDR Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE##GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
system defined		RECID##GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY		discarded	
TERMINAL.RMV		discarded	Duplicate of HEADER info
OPERATOR.RMV		discarded	Duplicate of HEADER info
TENDER.RMV		TENDERTY	short integer Table H-1 on page H-1
DATETIME.RMV		discarded	Duplicate of HEADER info
0		AMTTENDE	decimal(15,0)
		CUSTOMER	char(24)
0		STATUS	short integer
0		AMTTENDCUR	decimal(15,0)
		USRDATA	char(26)

String '28' Void Coupons String to QAZCFG TAX

String 28 captures tax information for every coupon offered as tender during a sales transaction. This string appears after STRING 14 and is inserted into the QAZCFG TAX table.

Source Information		QAZCFG TAX Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFGHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFGHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFGHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFGHDR		TDATE	date (Key)
Foreign Key Reference QAZCFGHDR		TTIME	time (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY		discarded	
1	AMTTAXA > 0	TAXTYPE	short integer
2	AMTTAXB > 0	TAXTYPE	short integer
3	AMTTAXC > 0	TAXTYPE	short integer
4	AMTTAXD > 0	TAXTYPE	short integer
1	REDTAXA > 0	TAXTYPE	short integer
2	REDTAXB > 0	TAXTYPE	short integer
3	REDTAXC > 0	TAXTYPE	short integer
4	REDTAXD > 0	TAXTYPE	short integer
AMTTAXA	TaxpType=(1)	AMT	decimal(15,0)
AMTTAXB	TaxpType=(2)	AMT	decimal(15,0)
AMTTAXC	TaxpType=(3)	AMT	decimal(15,0)
AMTTAXD	TaxpType=(4)	AMT	decimal(15,0)
REDTAXA	TaxpType=(1)	(-)AMT	decimal(15,0)
REDTAXB	TaxpType=(2)	(-)AMT	decimal(15,0)
REDTAXC	TaxpType=(3)	(-)AMT	decimal(15,0)
REDTAXD	TaxpType=(4)	(-)AMT	decimal(15,0)
INDICAT0		discarded	

General Sales Cross Reference

Appendix G. Supermarket Application Cross Reference

This appendix maps the **Supermarket Application** to the QUSRPOSSA Collection's predefined table definitions installed with RAIS option 11. This mapping is also shown by the QZCFSADB template defined that is included as part of the option 11 installation.

The mapping is as follows:

<i>Table G-1. String Description.</i>			
Source Information		Target Information	
Field	Bit Number	Column Name	Column Type
FFFFFFF	BBBBBBB	TTTTTTT	CCCCCC

Where:

- Field** The name of the field from which the row is derived.
- Bit Number** The number of the bit from which the row is derived.
- Table** The name of the table which now contains the data.
- Column** The name of the column in the table which now contains the data.

Notes:

1. The definition table may not refer to all of the fields in every case.
2. There may not be mapping in all cases.
3. This reference information can be found in the programming guide for each application.

EAMTRAN* - SA Transaction Summary Log

This appendix is a cross reference between detail files defined by the 4680 Supermarket Application and associated features and the QUSRPOSSA relational database tables.

Reference:

- SC30-3634 4680-4690 Supermarket Application: Programming Guide
- SC30-3513 4680 Supermarket Application: Electronic Funds Transfer (EFT)
- SC30-3461 4680 Price Management Feature: User's Guide

String '00' Transaction Log Header to QAZCFSHDR

Source Information		QAZCFSHDR Information	
Field	Bit Number	Column Name	Column Type
system defined		STORE#GEN	char(6) (Key)
TYPE		discarded	
TERMINAL		TERMINAL	char(4) (Key)
TRANSNUM		TRANSNUM	integer (Key)
DATE TIME		TDATE TIME	datetime (Key)
0		KEYEXTEND	short integer (Key)
TRANTYPE		TRANTYPE	short integer
NUMSTRIN		NUMSTRIN	short integer
OPERATOR		OPERATOR	char(10)
PASSWORD	To discard, set the template Retail data type to drop(6).	PASSWORD	integer
GROSSPOS		GROSSPOS	decimal(15,0)
GROSSNEG		GROSSNEG	decimal(15,0)
RINGTIME		discarded	
TENDERTI		discarded	
SPECIAL		discarded	
INACTIVE		discarded	
0		NETCASH.	decimal(15,0)
0		NETNONCASH	decimal(15,0)
0		NETCASHCUR	decimal(15,0)
INDICAT1	Bit0 X'80000000' Must be Zero	INDICAT1	char(1)
INDICAT1	Bit1 X'40000000' TOF-recovered transaction	IN1TOF	numeric(1,0)
INDICAT1	Bits 2-4 Reserved	IN1RSRV	char(3)
INDICAT1	Bits 5-9 User	IN1USER	char(5)
INDICAT1	Bit10 X'00200000' Gross Positive field is negative	IN1NGRSPOS	numeric(1,0)
INDICAT1	Bit11 X'00100000' Gross Negative field is negative	IN1NGRSNEG	numeric(1,0)
INDICAT1	Bit12 X'00080000' Additional Records for this transaction	IN1ADLRCDS	numeric(1,0)
INDICAT1	Bit13 X'00040000' Not first record for this transaction	IN1N1STRCD	numeric(1,0)
INDICAT1	Bit 14 X'00020000' Sign-off record contains till change string due to LAN failure	IN1SOTLCHG	numeric(1,0)
INDICAT1	Bit 15 X'00010000' Terminal initialized before or during this transaction	IN1TRMINIT	numeric(1,0)
INDICAT1	Bit 16 X'00008000' Reserved	IN1RSRV2	numeric(1,0)
INDICAT1	Bit 17 X'00004000' Current or previous period accounting file update - used by extended accounting tender by variety	IN1ACCNT	numeric(1,0)
INDICAT1	Bit 18 X'00002000' Transaction transferred	IN1TRNTFR	numeric(1,0)
INDICAT1	Bit 19 X'00001000' Transaction monitored	IN1TRNMNT	numeric(1,0)

Supermarket Application Cross Reference

Source Information		QAZCFSHDR Information	
INDICAT1	Bit 20 X'00000800' No-sale tender removal before or during this transaction	IN1TNDRRMV	numeric(1,0)
INDICAT1	Bit 21 X'00000400' No-sale till contents report before this transaction	IN1TILLRPT	numeric(1,0)
INDICAT1	Bit 22 X'00000200' No-sale till exchange before this transaction	IN1TILLXCH	numeric(1,0)
INDICAT1	Bit 23 X'00000100' No-sale tender verified before this transaction	IN1TNDRVFY	numeric(1,0)
INDICAT1	Bit 24 X'00000080' New password entered with the operator sign-on	IN1NEWPWD	numeric(1,0)
INDICAT1	Bit 25 X'00000040' Operator sign-on prior to the transaction	IN1SIGNON	numeric(1,0)
INDICAT1	Bit 26 X'00000020' Tender rejected in transaction	IN1TNDRRJT	numeric(1,0)
INDICAT1	Bit 27 X'00000010' Sign-off is not a true sign-off; NRTs logged	IN1NOTSGNO	numeric(1,0)
INDICAT1	Bit 28 X'00000008' Data entry made with this transaction (in next string)	IN1DATAENT	numeric(1,0)
INDICAT1	Bit 29 X'00000004' Cash drawer opened by no-sale or non-sales procedure before or during this transaction	IN1CSHRWR	numeric(1,0)
INDICAT1	Bit 30 X'00000002' Special sign-off before or during this transaction	IN1SPCLSOF	numeric(1,0)
INDICAT1	Bit 31 X'00000001' Terminal accountability	IN1TRMACCT	numeric(1,0)
<p>Note: The Primary Keyed field order in the QAZCFSHDR file is:</p> <p>STORE##GEN TERMINAL TRANSNUM TDATETIME KEYEXTEND</p>			

String '00' Transaction Log Header to QAZCFSTTIM

Source Information		QAZCFSTTIM Information	
Field	Bit Number	Column Name	Column Type
system defined		STORE#GEN	char(6)
TYPE		discarded	
TERMINAL		TERMINAL	char(4)
TRANSNUM		TRANSNUM	integer
DATE TIME		TDATE TIME	datetime
	0	KEYEXTEND	short integer
TRANTYPE		discarded	
NUMSTRIN		discarded	
OPERATOR		OPERATOR	char(10)
PASSWORD		discarded	
GROSSPOS		discarded	
GROSSNEG		discarded	
RINGTIME		RINGTIME	integer
TENDERTI		TENDTIME	integer
SPECIAL		SPECIAL	integer
INACTIVE		INACTIME	integer
INDICAT1		discarded	
<p>Note: The Primary Keyed field order in the QAZCFSTTIM file is: NONE</p>			

Supermarket Application Cross Reference

String '01' Merchandise / Non Merchandise String to QAZCFMRCH

This string is included for each merchandise/nonmerchandise line item entry. This string is contained in customer checkout transaction types only.

Source Information		QAZCFMRCH Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATETIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	
ITEMCODE		ITEMCODE	char(26)
XPIRCE		AMTITEM	decimal(16,0)
DEPARTME		DEPARTME	char(6)
FAMILYNU		FAMILY	char(6)
INDICAT1	Bit 7 - X'8000' Cannot be used	INDICAT1	numeric(1,0)
INDICAT1	Bit 6 - X'4000' Weight item	IN1WEIGHT	numeric(1,0)
INDICAT1	Bit 5 - X'2000' Entered price used	IN1ENTPRC	numeric(1,0)
INDICAT1	Bit 4 - X'1000' Price required	IN1PRCRQD	numeric(1,0)
INDICAT1	Bit 3 - X'0800' Log all occurrences	IN1LOGALL	numeric(1,0)
INDICAT1	Bit 2 - X'0400' Log item as exception	IN1LOGEXCP	numeric(1,0)
INDICAT1	Bit 1 - X'0200' Item code from alias record	IN1FRALIAS	numeric(1,0)
INDICAT1	Bit 0 - X'0100' No item movement kept	IN1NITMMVM	numeric(1,0)
INDICAT1	Bit 15 - X'0080' Tax plan A applies to this item	IN1TAXA	numeric(1,0)
INDICAT1	Bit 14 - X'0040' Tax plan B applies to this item	IN1TAXB	numeric(1,0)
INDICAT1	Bit 13 - X'0020' Tax plan C applies to this item	IN1TAXC	numeric(1,0)
INDICAT1	Bit 12 - X'0010' Tax plan D applies to this item	IN1TAXD	numeric(1,0)
INDICAT1	Bit 11 - X'0008' Food stamp item	IN1FOODSTP	numeric(1,0)
INDICAT1	Bit 10 - X'0004' Trading stamp item	IN1TRDSTMP	numeric(1,0)
INDICAT1	Bit 9 - X'0002' Non-discountable item	IN1NONDSC	numeric(1,0)
INDICAT1	Bit 8 - X'0001' Coupon multiplication allowed	IN1CMLTALW	numeric(1,0)
INDICAT2	Bit 7 - X'8000' Cannot be used	INDICAT2	numeric(1,0)

Source Information		QAZCFSMRCH Information	
INDICAT2	Bit 6 - X'4000' Item requires override	IN2RQOVER	numeric(1,0)
INDICAT2	Bit 5 - X'2000' Data entry with item	IN2DATAENT	numeric(1,0)
INDICAT2	Bit 4 - X'1000' Multipriced item	IN2MLTIPRC	numeric(1,0)
INDICAT2	Bit 3 - X'0800' Weight or quantity entered	IN2QTYENT	numeric(1,0)
INDICAT2	Bit 2 - X'0400' Store coupon created by coupon multiplier	IN2STRCPCR	numeric(1,0)
INDICAT2	Bit 1 - X'0200' Tax key pressed	IN2TAXKEY	numeric(1,0)
INDICAT2	Bit 0 - X'0100' Food stamp key pressed	IN2FOODSTK	numeric(1,0)
INDICAT2	Bit 15 - X'0080' Cancel key pressed	IN2CNCLKEY	numeric(1,0)
INDICAT2	Bit 14 - X'0040' Refund key pressed	IN2RFNDKEY	numeric(1,0)
INDICAT2	Bit 13 - X'0020' Store coupon key pressed	IN2STRCPNK	numeric(1,0)
INDICAT2	Bit 12 - X'0010' Manufacturers coupon key hit	IN2MNFCPNK	numeric(1,0)
INDICAT2	Bit 11 - X'0008' Deposit key pressed	IN2DPSTKEY	numeric(1,0)
INDICAT2	Bit 10 - X'0004' Negative price due to deal	IN2NPRDEAL	numeric(1,0)
INDICAT2	Bit 9 - X'0002' Extension follows this string	IN2EXTENSN	numeric(1,0)
INDICAT2	Bit 8 - X'0001' Reserved	IN2RSRV	numeric(1,0)
INDICAT3	T<6, T>7	INDICAT3	decimal(2,0)

Supermarket Application Cross Reference

String '01' Merchandise / Non Merchandise String to QAZCFSTNDR

This string is included for each Manufacturer and Store Coupon item entry for sales transactions only.

Source Information		QAZCFSTNDR Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATETIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	
Manufacturer coupon = 210, Store coupon = 200		TENDERTY	smallint
ITEMCODE		ITEMCODE	char(26)
XPIRCE		AMTTENDE	decimal(16,0)
DEPARTME		discrded	
FAMILYNU		FAMILYNU	decinal(3,0)
INDICAT1		discarded	
INDICAT2	Bits 0-7	INDICAT2	
INDICAT2	Bit 15 - X'0080' Cancel key pressed	VOIDED	
INDICAT2	Bits 8-14	IN2RSRV	
INDICAT3	T = 6 Manufacturer coupon, T = 7 Store coupon	COUPONCD	char

String '02' Merchandise / Non Merchandise String Extension to QAZCFSMRCX

This string is written for each item entry that is multipriced, had a price override, or had a weight or quantity entered. It always follows the item entry string to which it applies..

The RECID##GEN column is used to maintain the order of items within a transaction. GSA and SA applications ensure the Item quantity, Item price override, or Item weight extension string immediately follows the Merchandise string in the transaction log. Therefore the RECID##GEN column for the QAZCFSMRCX table is one greater than the RECID##GEN value for the associated QAZCFSMRCH record.

Source Information		QAZCFSMRCX Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE##GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATETIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID##GEN	short integer (Key) Incremented for each record in a transaction
STRINGTY		discarded	
MPGROUP		EXTGROUP	short integer
DEALQUAN		DEALQUAN	decimal(10,0)
PRICEMETHD		PRICEMETHD	decimal(2,0)
SALEQUAN		SALEQUAN	decimal(10,0)
SALEPRIC	This field is the concatenated Deal Price and Unit price and requires the Item Record pricing method to parse.	SALEPRIC	decimal(15,0)
QTYORWGT	Number of Items, or weight	QUANTITY	decimal(10,2)
INDICAT1	Bit 7 - X'80' Cannot be used	INDICAT1	char
INDICAT1	Bit 5 - X'20' Quantity keyed	IN1QUANKEY	numeric(1,0)
INDICAT1	Bit 4 - X'10' Tare code keyed	IN1TARECOD	numeric(1,0)
INDICAT1	Bit 3 - X'08' Weight from scale	IN1WGHTSCL	numeric(1,0)
INDICAT1	Bit 2 - X'04' Quantity required	IN1QUANRQD	numeric(1,0)
INDICAT1	Bit 1 - X'02' Quantity through merging of consecutive identical entries.	IN1QUANCLC	numeric(1,0)
INDICAT1	Bit 0 - X'01' Coupon enlarged quantity	IN1QUANCPN	numeric(1,0)

Supermarket Application Cross Reference

String '03' and '04' Discount/void Discount String to QAZCFSDISC

This string is included for each transaction discount and voided discount entry. These strings are contained in customer checkout transaction records only.

Source Information		QAZCFSDISC Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATEIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	
DISGROUP		DISGROUP	decimal(15,0)
PERDISCO	Rate	PERDISCO	decimal(5,2)
AMTDISCO		AMTDISCO	decimal(15,0)
TAXEXEMP		TAXEXEMP	decimal(15,0)
	For string type = 3, indvoided = 0. For string type 4, indvoided = 1.	INDVOIDED	numeric(1,0)

String '05' and '06' Tender/Tender Correction String to QAZCFSTNDR

These strings are contained in customer checkout, tender cashing, and tender exchange transaction records.

Source Information		QAZCFSTNDR Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATETIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	
TENDTYPE		TENDERTY	short integer
AMTTENDE		AMTTENDE	decimal(15,0)
AMTTNFEE		AMTTNFEE	decimal(15,0)
CUSTOMER		CUSTOMER	char(24)
STATUS		STATUS	short integer
	For string type = 5, voided = 0. For string type 6, voided = 1.	VOIDED	numeric(1,0)

String '07' and '08' Tax/Tax Refund String to QAZCFSTAX

These strings are contained in customer checkout transaction records only.

Source Information		QAZCFSTAX Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATETIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	
1	AMTTAXA > 0	TAXTYPE	short integer
2	AMTTAXB > 0	TAXTYPE	short integer
3	AMTTAXC > 0	TAXTYPE	short integer
4	AMTTAXD > 0	TAXTYPE	short integer
AMTTAXA	TAXTYPE=(1)	AMT	decimal(15,0)
AMTTAXB	TAXTYPE=(2)	AMT	decimal(15,0)
AMTTAXC	TAXTYPE=(3)	AMT	decimal(15,0)
AMTTAXD	TAXTYPE=(4)	AMT	decimal(15,0)
AMTSALEA	TAXTYPE=(1)	AMTSALE	decimal(15,0)
AMTSALEB	TAXTYPE=(2)	AMTSALE	decimal(15,0)
AMTSALEC	TAXTYPE=(3)	AMTSALE	decimal(15,0)
AMTSALED	TAXTYPE=(4)	AMTSALE	decimal(15,0)
	For string type = 7, negative = 0 For string type 8, negative = 1.	NEGATIVE	numeric(1,0)

String '09' Food Stamp Change String to QAZCFSTNDR

This string is included for each transaction Food Stamp Change entry for sales transactions. There is an entry for all change returned.

Source Information		QAZCFSTNDR Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE##GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATETIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID##GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	
TENDTYPE		TENDERTY	short integer
AMTTENDE		AMTTENDE	decimal(15,0)
0		AMTTNFEE	decimal(15,0)
		CUSTOMER	char(24)
		STATUS	short integer
1		NEGATIVE	numeric(1,0)
0		VOIDED	numeric(1,0)

String '10' Manager Override String to QAZCFSMGRO

This string appears in any transaction record which required a manager override. It normally follows the string which required the override.

Source Information		QAZCFSMGRO Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE##GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATETIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID##GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	
OVERRIDE		OVERRIDE	char(7)
REASON		REASON	integer

String '11' Transaction Data Entry String to QAZCFSDTAE

This string can be present in any transaction record. It can follow a transaction header, an item entry, or a tender in a customer checkout transaction where it contains data entered for the transaction, item, or tender. It can also be entered as the last string in any transaction from a user exit.

Source Information		QAZCFSDTAE Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE##GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATETIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID##GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	
DATA		DATA1	char(10)
DATA		DATA2	char(10)
DATA		DATA3	char(10)
DATA		DATA4	char(10)
DATA		DATA5	char(10)
DATA		DATA6	char(10)

String '12' Transaction Price Change String to QAZCFSPRCH

This string is contained in a price change transaction record only.

Source Information		QAZCFSPRCH Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATETIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	
ITEMCODE		ITEMCODE	char(26)
NEWSQUAN		NEWSQUAN	decimal(10,2)
NEWPRICE		NEWPRICE	decimal(15,0)
OLDSQUAN		OLDSQUAN	decimal(10,2)
OLDPRICE		OLDPRICE	decimal(15,0)

String '13' Transaction Cash Drawer String to QAZCFSCSDW

This string may be contained in a Loan, Pickup, Tender Count, Standalone, or Sign-off transaction record. Amounts are positive if they represent money into the till and are negative if they represent money out of the till. Thus, pickup amounts are normally negative and loan amounts are normally positive.

There will be 1 record created for each entry in the Cash drawer String.

Source Information		QAZCFSCSDW Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE##GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATETIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID##GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	
NUMLOANS		discarded	
AMTLOANS		discarded	
NUMPKUPS		discarded	
AMTPKUPS		discarded	
TENDTYPx	x = 1 - 8	TENDER	short integer
AMTTENDx	x = 1 - 8	AMTTEND	decimal(15,0)
NEGATIVE	'95' delimiter for negative tenders	discarded	
Set to 1 for all entries after '95' delimiter for negative tenders.		NEGATIVE	numeric(1,0)

String '14' Transaction Item Record Change String to QAZCFSIRCH

This string is contained in a price change transaction only. It is used when more than just the price is changed in an item record.

The item record data is a string containing all data from the item record except for the descriptor. Item record fields are delimited by colons and all integer fields are expressed as ASCII numerics. The string OLDDATA and NEWDATA are separated by three equal signs, which are converted to a comma before being written to the Exception Log.

Source Information		QAZCFSIRCH Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATETIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	
OLDDATA		OLDDATA	char(50)
NEWDATA		NEWDATA	char(50)

String '15' Transaction Trading Stamps String to QAZCFSSTMP

This string is contained in a customer checkout transaction record when trading stamps are given to the customer.

Source Information		QAZCFSSTMP Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE##GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATEIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID##GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	
STAMPS		STAMPS	integer

Supermarket Application Cross Reference

String '20' Transaction Exception Log String

This string is included for each transaction Transaction Log for all transactions.

Not Processed

String '21' Store Closing String to QAZCFSHDR

This string comprises a record by itself and marks the closing point for a set of files.

Both a Header entry and a close store entry entry are created.

Source Information		QAZCFSHDR Information	
Field	Bit Number	Column Name	Column Type
system defined		STORE#GEN	char(6) (Key)
TYPE		discarded	
0		TERMINAL	char(4) (Key)
0		TRANSNUM	integer (Key)
DATE TIME		TDATE TIME	datetime (Key)
0		KEYEXTEND	short integer (Key)
1011		TRANTYPE	short integer
0		NUMSTRIN	short integer
0		OPERATOR	char(10)
	To discard, set the template Retail data type to drop(6).	PASSWORD	integer
0		GROSSPOS	decimal(15,0)
0		GROSSNEG	decimal(15,0)
0		NETCASH.	decimal(15,0)
0		NETNONCASH	decimal(15,0)
0		NETCASHCUR	decimal(15,0)

String '21' Store Closing String to QAZCFSCLOS

This string is included for each Store Close entry for sales transactions.

Source Information		QAZCFSCLOS Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATE TIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	

Table G-18 (Page 2 of 2). String '21' Store Close String.

Source Information		QAZCFSCLOS Information	
DATETIME		discarded	
INDICAT1		CLOSTYPE	short integer
INDICAT2	Bit5-7 X'80' Reserved	INDICAT2	char(3)
INDICAT2	Bit4 X'10' Close tender listing log	IN2TNDLOG	numeric(1,0)
INDICAT2	Bit3 X'08' Close terminal productivity log	IN2TRMLOG	numeric(1,0)
INDICAT2	Bit2 X'04' Close exception log	IN2EXCPLOG	numeric(1,0)
INDICAT2	Bit1 X'02' Close item movement totals	IN2MVMTTTL	numeric(1,0)
INDICAT2	Bit0 X'01' Close customer account status	IN2CSTACTT	numeric(1,0)

String '98' Operator in Training String to QAZCFSOTRN

This string comprises a record by itself and is written at the end of every transaction while in operator training mode.

Source Information		QAZCFSOTRN Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATETIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	
1		INDTRAIN	

String '99' User Data String to QAZCFSUSER

This string is included for each User Data entry.

Source Information		QAZCFSOTRN Information	
Field	Bit Number	Column Name	Column Type
Foreign Key Reference QAZCFSHDR		STORE#GEN	char(6) (Key)
Foreign Key Reference QAZCFSHDR		TERMINAL	char(4) (Key)
Foreign Key Reference QAZCFSHDR		TRANSNUM	integer (Key)
Foreign Key Reference QAZCFSHDR		TDATETIME	datetime (Key)
Foreign Key Reference QAZCFSHDR		KEYEXTEND	short integer (Key)
system defined		RECID#GEN	short integer (Key) Incremented for each record in a transaction
TYPE		discarded	
USRDATA		USRDATA	char(99)

Appendix H. Transaction Detail Type Definitions

These tables provide a common definition of values associated with certain selection types of data. These tables are **NOT** a part of the physical database.

Valid Types of Tenders

- Table 6-7 on page 6-9, TENDER.TndrType
- Table 6-8 on page 6-10, TENDERCHG.TndrType
- Table 6-18 on page 6-21, ISSUETNDR.TndrType
- Table 6-6 on page 6-8, DISCVIEW.TndrType
- Table 6-14 on page 6-17, ITEMDISCVIEW.TndrType
- Table 6-25 on page 6-28, PICKUPS.TndrType
- Table 6-26 on page 6-29, PAYOUTS.TndrType
- Table 6-27 on page 6-30, PAYINS.TndrType

Design note

Supermarket Application personalization options support tender types to be added to support the 4680 supermarket for tender varieties (e.g. 011 = Cash tender Variety 1, 012 = Cash tender Variety 2).

General Sales Application has personalization options where the user can define 35 different values for the tender types. **General Sales Application** Version two extended the tender types to support foreign currencies. The customer personalizes the tender types to match the currencies they support. The QUSRPOSGS/QAZCFGAPMP file can be modified to capture these foreign currency tender types and convert them to CurrType values.

Foreign currency is not valid for tender types 0, 15 and GSA charge tenders 6 through 9.

Table H-1 (Page 1 of 2). Valid Types of Tenders.

RESERVED TYPE/RULE VALUES:

- 00,000 - 04,999 = IBM Values
- 05,000 - 09,999 = Business Partners
- 10,000 - 32,000 = Customer Defined

Type	Description	SA Default values	GSA Default values	RA Default values
010	cash	1v	15	60
020	check	2v	0	61
030	foodstamp	3v	--	--
040	Miscellaneous tender1	4v	14	68
050	Miscellaneous tender2	5v	--	69
060	Miscellaneous tender3	6v	--	--
070	gift certificate	--	1	64
080	due bill	--	2	--
090	travelers check	--	3	62
200	store coupon	--	4	67
210	manufactures coupon	--	5	--
220	electronic store coupon	--	--	--
230	electronic manufactures coupon	--	--	--
300	charge A	--	6	--
310	charge B	--	7	--
320	charge C	--	8	--
330	charge D	--	9	--
340	Visa card	--	10	--

Supermarket Application Cross Reference

Table H-1 (Page 2 of 2). Valid Types of Tenders.

RESERVED TYPE/RULE VALUES:

00,000 - 04,999 = IBM Values

05,000 - 09,999 = Business Partners

10,000 - 32,000 = Customer Defined

Type	Description	SA Default values	GSA Default values	RA Default values
350	Master card	--	11	--
360	American Express card	--	12	--
370	Diners Club card	--	13	--
380	Discover card	--	--	--
500	EFT	--	--	--
1000	House Charge	--	--	63
1010	Store Gift Certificate	--	--	63
1020	Mall Gift Certificate	--	--	64
1030	Store Credit	--	--	65
1040	Coupon	--	--	66
1050	Credit Card 1	--	--	70
1060	Credit Card 2	--	--	71
1070	Credit Card 3	--	--	72
1080	Credit Card 4	--	--	73
1090	Credit Card 5	--	--	74
1100	Credit Card 5	--	--	75
10001	Foreign currency tender type	--	16	101
10002	Foreign currency tender type	--	17	102
10003	Foreign currency tender type	--	18	103
10004	Foreign currency tender type	--	19	104
10005	Foreign currency tender type	--	20	105
10006	Foreign currency tender type	--	21	106
10007	Foreign currency tender type	--	22	107
10008	Foreign currency tender type	--	23	108
10009	Foreign currency tender type	--	24	109
10010	Foreign currency tender type	--	25	110
10011	Foreign currency tender type	--	26	111
10012	Foreign currency tender type	--	27	112
10013	Foreign currency tender type	--	28	--
10014	Foreign currency tender type	--	29	--
10015	Foreign currency tender type	--	30	--
10016	Foreign currency tender type	--	31	--
10017	Foreign currency tender type	--	32	--
10018	Foreign currency tender type	--	33	--
10019	Foreign currency tender type	--	34	--
10020	Foreign currency tender type	--	35	--

Valid Types of Transaction Functions

This table contains the valid transaction functions.

Note: This table contains data for Table 6-1 on page 6-3 TRANAMOUNT.FuncType

Source: GSA, RAD and SA(Transaction types)

Table H-2 (Page 1 of 4). Valid Types of Transaction Functions.	
RESERVED TYPE VALUES:	
00,000 - 04,999 = IBM Values	
05,000 - 09,999 = Business Partners	
10,000 - 32,000 = Customer Defined	
Value	Description
----	Sales Type Transactions 0000 - 0999
0001	Checkout Transaction (Normal Sales transaction) Source: SA Checkout transaction type 00 Source: GSA Sales transaction type 1 Source: RAD Normal transaction type 0
0002	Employee Discount Transaction Retail Application Note that RAD discount type of transactions will make entries into the ITEMDISC table.
0003	Club 1 discount
0004	Club 2 discount
0005	Record miscellaneous transaction
0006	Voided checkout transaction Design Note: GSA reports a transaction is Voided in a INDICAT0.HDR bit 3 where supermarket changes the transaction type from 00 (customer checkout) to 16 (Voided Transaction). Design Note: RAD reports a transaction is Voided in the header string CANCEL STATUS field.
0007	Cash Special Transaction Source: GSA Sales transaction type 2
0008	Cash Document Transaction Source: GSA Sales transaction type 3
0009	Cash On Delivery (COD) transaction Source: GSA Sales transaction type 4
0010	Layaway Transaction Source: GSA Sales transaction type 5 Source: RAD Transaction type 2
0011	Charge - Credit plan A, B, C, or D General Sales Application Refer to the TndrType in TNDR for the appropriate charge card. Source: GSA Sales transaction type 6, 7, 8 and 9

Supermarket Application Cross Reference

Value	Description
<p><i>Table H-2 (Page 2 of 4). Valid Types of Transaction Functions.</i></p> <p>RESERVED TYPE VALUES:</p> <p>00,000 - 04,999 = IBM Values 05,000 - 09,999 = Business Partners 10,000 - 32,000 = Customer Defined</p>	
0012	Special Order
0013	Job Accounting <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> <p>&smabox.</p> <p>SMA currently used Charge Plan D for Job Accounting.</p> </div>
0014	No-sale transaction <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> <p>RAD</p> <p>This function occurs for transaction type 5.</p> </div>
0015	Operator Sign-on Attempt <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> <p>GSA</p> <p>This function occurs for invalid sign-on attempts only (String 19).</p> </div>
----	Accounting Functions 1000 - 1999
1001	Tender Cashing Source: SA Checkout transaction type 01 Source: RAD Check cash transaction type 09
1002	Tender Exchange Source: SA Checkout transaction type 02
1003	Cashier Loan Source: SA Checkout transaction type 03 Source: GSA Non-Sales transaction type 50 Source: RAD Payin transaction type 06
1004	Cashier Pickup Source: SA Checkout transaction type 04 Source: GSA Non-Sales transaction type 51 Source: RAD Cash Payout transaction type 08
1005	Tender Count Source: SA Checkout transaction type 10 Source: GSA Non-Sales transaction type 52
1006	Totals readout and reset transaction Source: GSA Non-sales Transaction type 53, Option 9 - Header string only (Totals are NOT reset) Source: RAD Non-sales Transaction type 14 option 2
1007	Final totals readout and reset transaction Source: GSA Non-Sales Transaction type 53 Option 1 or 2 - Header string and string 3 Source: RAD Totals Transaction type 14 option 1
1008	Check/Tender Removal transaction Source: GSA Non-Sales transaction type 63 Source: RAD Pickup transaction type 07
1009	Void Previous transaction Source: GSA Non-Sales transaction type 60 Source: RAD Non-Sales transaction type 40
1010	Void Previous transaction by line item Source: GSA Non-Sales transaction type 61

Table H-2 (Page 3 of 4). Valid Types of Transaction Functions.	
RESERVED TYPE VALUES: 00,000 - 04,999 = IBM Values 05,000 - 09,999 = Business Partners 10,000 - 32,000 = Customer Defined	
Value	Description
----	Request for a Report 2000 - 2999
2001	Price verify (ONLY) Source: Supermarket Price verify/change (06) not followed by any string. Source: DRG Price Check function.
2002	Tender Listing Source: SA Checkout transaction type 05
2003	Report Department totals Source: GSA Non-Sales transaction type 54
2004	Report Item Movement Source: GSA Non-Sales transaction type 55
2005	Report Check listing Source: GSA Non-Sales transaction type 62
2006	Report Directory, Terminal RAM disk
----	Data Maintenance Requests 3000 - 3999
3000	Price change transaction Source: Supermarket Price verify/change (06) followed by String 12 Source: GSA Non-Sales transaction type 56
3001	Item Record Change Source: Supermarket Price verify/change (06) followed by String 14
3002	Modify preset/department keys
3003	Change salesperson password
3004	Salesperson maintenance
3005	Security maintenance
3006	UPC coupon value code table maintenance
----	Workstation Control 4000 - 4100
4000	Training Session Source: SA Checkout transaction type 07 Source: GSA Non-Sales transaction type 59 Source: RAD Non-Sales transaction mode 02
4001	Terminal Transfer Source: SA Checkout transaction type 08
4002	Terminal Monitor Source: SA Checkout transaction type 09 Source: GSA Non-Sales transaction type 64
4003	Operator sign-off Source: SA Checkout transaction type 17 Source: RAD Checkout transaction type 04
4004	Standalone Session Source: GSA INDICAT0.HDR bit 7 Source: SA Checkout transaction type 18 Source: RAD Non_sales transaction mode 03
4005	Set transaction number Source: GSA Non-Sales transaction type 57

Supermarket Application Cross Reference

Value	Description
<p><i>Table H-2 (Page 4 of 4). Valid Types of Transaction Functions.</i></p> <p>RESERVED TYPE VALUES:</p> <p>00,000 - 04,999 = IBM Values 05,000 - 09,999 = Business Partners 10,000 - 32,000 = Customer Defined</p>	
4006	<p>Reentry transaction (header only).</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p align="center">General Sales Application</p> <p>Check this one out in the TLOG. A-84 of GSA programmers guide states. A re-entry transaction header is written when transactions have not been re-entered in a re-entry session.</p> </div> <p>Source: GSA Non-Sales transaction type 58 Source: RAD Non-Sales transaction mode 04</p>
4007	<p>Terminal program load Source: GSA Non-Sales transaction type 65</p>
4008	<p>Controller sign-on</p>
4009	<p>Retrieval of a suspended transaction Source: GSA Sales transaction type 1 and INDICAT0.HDR bit 4</p>
4010	<p>Suspended transaction Source: GSA Sales transaction INDICAT0.HDR bit 5</p>
4050	<p>Rain check transaction Source: RAD Sales transaction type 13</p>
4051	<p>Operator Sign-on Source: GSA Non-Sales transaction type 03</p>

Types of Data Entry

This table will contain an identifier for the valid types of user defined data strings.

Note: This table contains data for

<p><i>Table H-3. Valid transaction data entry types.</i></p> <p><i>RESERVED TYPE VALUES:</i></p> <p><i>00,000 - 04,999 = IBM Values</i></p> <p><i>05,000 - 09,999 = Business Partners</i></p> <p><i>10,000 - 32,000 = Customer Defined</i></p>	
Value	Description
01	Supermarket Field1
02	Supermarket Field2
03	Supermarket Field3
04	Supermarket Field4
05	Supermarket Field5
06	Supermarket Field6

Supermarket Application Cross Reference

EFT - Valid EFT Actions

Note: This table contains data for -- Table 'EFT' unknown --, EFT.TotalType.

Type	Description	SA Default values	GSA Default values	RA Default values
<p><i>Table H-4. Valid EFT Actions.</i></p> <p><i>RESERVED TYPE/RULE VALUES:</i></p> <p><i>00,000 - 04,999 = IBM Values</i></p> <p><i>05,000 - 09,999 = Business Partners</i></p> <p><i>10,000 - 32,000 = Customer Defined</i></p>				
01	Online approval	String 16 X'1'	String 21 x'1'	
02	Online rejected	String 16 X'3'	String 21 x'3'	
03	Online void approved	String 16 X'5'	String 21 x'5'	
04	Stand-in approved	String 16 X'7'	String 21 x'7'	
05	Stand-in rejected	String 16 X'9'	String 21 x'9'	
06	Stand-in void approved	String 16 X'11'	String 21 x'11'	
07	Online request approved in training	String 16 X'13'	N/A	
08	Online request rejected in tra	String 16 X'15'	N/A	
09	Void in training	String 16 X'17'	N/A	
10	Online refund approved	String 16 X'19'	N/A	
11	Online refund rejected	String 16 X'21'	N/A	
12	Stand-in refund approved	String 16 X'23'	N/A	
13	Stand-in refund rejected	String 16 X'25'	N/A	
14	EFT Accepted	String 30 X'19'	String 23 x'15'	
15	EFT Rejected	String 30 X'21'	String 23 x'17'	
16	Void rejected	String 30 X'23'	String 23 x'21'	
17	Refund approved	String 30 X'39'	N/A	
18	Online count of Time-out Reversals	String 30 X'40'	String 23 x'40'	
19	Refund rejected	String 30 X'41'	N/A	
20	Void approved	String 30 X'60'	String 23 x'19'	
21	Count of Time-out Reversals in reconciliation of stand-in	String 30 X'71'	String 23 x'41'	
22	Stand-in void rejected	N/A	String 21 x'13'	
23	Reentry of a request	N/A	String 21 x'29'	
24	Reentry of a void	N/A	String 21 x'30'	

Appendix I. Authorization

This appendix provides information and guidance for setting authorization to update and access data.

Brief Security Overview

The AS/400 has object authorization and data authorization. Object authority is broken into operational, management, existence, alter, and reference authorities. Data authorization is broken into read, add, update, delete, and execute authorities. Each of these authorities is explained in great detail in the OS/400 Security Reference manual.

The AS/400 provides a general authorization of *USE which grants a profile or authorization list operational authority and read and execute data authority. For users wanting to update objects, there is an authorization of *CHANGE which grants operational authority and all of the data authorizations. In addition, users can be granted a combination of these authorities as needed.

QRAIS User Profile Authorizations

Calling User Exit Program

The QRAIS profile needs to have *USE authority granted on the exit program to successfully run it.

Non-relational data conversion

The QRAIS profile needs to have *ADD to the libraries specified on the retail file and AS/400 file field in the template.

If the file already exists in the library, QRAIS needs *OBJMGT or *OBJALTER authority to the files.

Relational Database data conversion

The QRAIS profile needs to have *W access to database tables specified in the template record entries and *X access to the database collection.

Note that the QRAIS profile owns the QUSRPOSGS collection and associated tables. Any new tables would require access to be granted for the polling application to insert data into them.

IFS data access

The QRAIS profile needs to have *X authority to each directory in the path preceding the file and *R authority to the file to be able to read.

Data Access User Authorizations

For users to access the relational data, they must be explicitly given the authority to do that. For users needing to look at the data only, *USE authority should be granted to the tables and views. For updating, the users should be given *CHANGE authority.

Granting authorizations

To grant authority to users, the GRTOBJAUT command should be used. This command can be used to grant to a specific user profile or to an authorization list which can have a group of user profiles associated with it. Refer to the CL reference for more information on the GRTOBJAUT command.

Provided Authorization Lists

The RAIS product comes with 2 authorization lists. RAISCFG and RAISDB. The RAISCFG authorization list has all configuration files per option installed and would be used to grant authority for configuration type operations.

The RAISDB authorization list has all database files for the product database options installed. This is the physical files as well as the logical files if they are installed.

The following is a list of authorities needed to configure RAIS.

```
OBJALTER, OBJOPR, ADD, EXECUTE, READ, UPD
```

This would grant authorities so the user can work with templates and configure polling information.

```
ADDAUTL AUTL(RAISCFG) USER(SAM) AUT( OBJALTER OBJOPR ADD EXECUTE  
READ UPD)
```

For the database access, the user needs *USE authority to read and *CHANGE to modify data.

Granting for User Profile

To grant *USE authority to all of the database tables for user SAM:

```
GRTOBJAUT OBJ(QUSRPOSGS/ ALL) OBJTYPE( FILE) USER(SAM) AUT( USE)
```

This would give SAM *USE authority on all files in the collection

Granting for Authorization List

Create an authorization list:

```
CRTAUTL AUTL(DBFILES)
```

Add user profile and authorization level to list:

```
ADDAUTLE AUTL(DBFILES) USER(SAM) AUT( CHANGE)
```

Associate the files with the authorization list

```
GRTOBJAUT OBJ(QUSRPOSGS/ ALL) OBJTYPE( FILE) AUTL(DBFILES)
```

Using QRAIS as a group profile

The QRAIS user profile is given *ALL object authority to the RAISDB Authorization list, and *CHANGE object authority to the RAISCFG Authorization list. You can specify the QRAIS profile as group profile (GRPPRF on the CRTUSRPRF command) to give your users the appropriate authority to these lists.

Bibliography

This section lists publications that contain information about topics described or referred to in this guide. The books in this section are listed with their full title and order number, but when referred to in text, a shortened version of the title is used. Refer to the *Publications Reference*, SC41-4003, if you want a list of all books in the AS/400 library. The *Publications Reference* also has a description of the information contained in those books.

AS/400 Books

The following AS/400 books contain information you may need when you use the *Retail Application Interface Suite for AS/400* book:

Retail Communications Programming, SC41-3448. Provides information pertaining to retail communications.

ICF Programming, SC41-3442. Contains information about writing application programs that use the inter-system communications function (ICF).

SNA Upline Facility Programming, SC41-3446. Contains information about using the Systems Network Architecture (SNA) upline facility (SNUF) with the AS/400 system.

Communications Configuration, SC41-5401-00. Contains general configuration information, including detailed descriptions of network interface, line, controller, device, mode, and class-of-service descriptions, configuration lists, and connection lists.

COBOL/400 Reference, SC09-1813.

CL Reference, SC41-4722. Provides the systems programmer or application programmer with a description of the AS/400 system control language (CL).

Retail Host Books

Information on the IBM Advanced Data Communication for Stores (ADCS) Licensed Program 5735-XR2 can be found in the following book:

Advanced Data Communication for Stores (ADCS) Program Reference and Operations Guide, SH20-2406.

Retail Controller Books

The following books contain additional information on retail controllers that you may need when you use this guide:

IBM 3650 Programmable Store System: Programmer's Guide, GC30-3062

IBM 3680 Programmable Store System: Programmer's Guide, GC30-3066

IBM 4680 Chain Drug Sales Application: Programming Guide, SC30-3414

IBM 4680 General Sales Application: Programming Guide, SC30-3370

IBM 4680 Store Management Application: Programming Guide, SC30-3487

IBM 4680 Store System: Programming Guide, SC30-3357

IBM 4680 Supermarket Application: Programming Guide, SC30-3373

IBM 4680-4690 General Sales Application: Guide to Operations, SC30-3632

IBM 4680-4690 General Sales Application: Full Screen - Guide to Operations, SC30-3664

IBM 4684 Store Sales Application: Program Reference, SB11-8472

Retail Industry Programming Support Services: Host Communication Package Programmer's Guide, SC33-0650

4690 Communications Programming Reference, SC30-3582

4690 Planning and Configuration Guide, GC30-3600

4690 Programming Guide, SC30-3602

4690 User's Guide, SC30-3597

IBM 4680 SNA LU 6.2 Communications, GG24-3623

Index

Special Characters

- *SEARCH 4-34
- *STORE 2-2, 4-34

Numerics

- 3270 emulation 1-5
- 3650
 - store systems 1-7
- 3651
 - keyed files 2-7
 - sequential files 2-7
 - store systems 1-1
- 3680
 - store systems 1-7
- 3684
 - keyed files 2-7
 - sequential files 2-7
 - store systems 1-1
- 4684
 - keyed files 2-7
 - sequential files 2-7
 - store systems 1-1, 1-7
- 4690
 - keyed files 2-7
 - sequential files 2-7
 - store systems 1-1, 1-7

A

- about this book xv
- ACCTDEP Table 6-24
- ACCTFEE Table 6-25
- ACCTNEW Table 6-22
- AcctNum
 - ACCTDEP Table 6-24
 - ACCTFEE Table 6-25
 - ACCTNEW Table 6-22
 - ACCTPAY Table 6-23
 - LAYCAN Table 6-26
 - LAYRSTK Table 6-27
 - SPECIALSALES Table 6-33
- ACCTPAY Table 6-23
- AcctType
 - ACCTDEP Table 6-24
 - ACCTNEW Table 6-22
 - ACCTPAY Table 6-23
- ADCS (Advanced Data Communications for Stores)
 - Application Program Interface (API) 3-1
 - coding examples
 - COBOL/400 3-1, 3-3
 - RPG/400 3-1

ADCS (Advanced Data Communications for Stores) (continued)

- commands
 - CREA 1-1
 - DUMP 1-1
 - KEYD 1-1
 - LOAD 1-1
 - PURG 1-1
 - RED 1-1
 - SUP 1-1
 - ZERO 1-1
- definition 1-4
- emulation
 - adding a keyed record command 4-17
 - deleting a keyed record command 4-17, 4-21
 - menus 4-1
 - new host (new PSS) format files 4-3
 - PC format files 4-2
 - point-of-sale communications 4-1
 - point-of-sale file types 4-2
 - point-of-sale function 1-4
 - print format files 4-2
 - PSS format files 4-2
 - reading a keyed record command 4-17
 - replacing a keyed record command 4-17
- introduction 4-1
- Load command 4-10
- pass-through functions 1-3, 1-4
 - using ADCS emulation 4-1
- adding a keyed record
 - to a retail controller file 4-17
- adding host file cross-references 4-25
- Adjust
 - ACCTNEW Table 6-22
- Advanced Data Communications for Stores (ADCS)
 - See ADCS (Advanced Data Communications for Stores)
- ADXBTBL.286 - 4690 file table maintenance program 2-6
- ADXSrvTL.286 - 4690 TCP/IP server 2-5
- Amount
 - ACCTDEP Table 6-24
 - ACCTFEE Table 6-25
 - ACCTNEW Table 6-22
 - ACCTPAY Table 6-23
 - ITEMDEPOSIT Table 6-18
 - LAYCAN Table 6-26
 - LAYRSTK Table 6-27
 - PAYINS table 6-30
 - PAYOUTS table 6-29
 - PICKUPS table 6-28
 - TNDRCASH table 6-31
- AmtDisc
 - DISCOUNT Table 6-8

AmtDisc (continued)

ITEMDISCOUNT Table 6-17

MDSESALES table 6-15

AmtDue

TENDER table 6-9

AmtForeign

TENDER table 6-9

AmtPVoid

POSTVOID table 6-12

AmtSales

LAYMDSE table 6-16

MDSESALES table 6-15

SPECIALSALES Table 6-33

SPRSNSALES Table 6-32

AmtTend

TENDER table 6-9

AmtTendBas

TENDER table 6-9

AmtTendChg

TENDERCHG table 6-10

AmtTender

ISSUETNDR Table 6-21

AmtVoid

ITEMVOID table 6-19

API (application program interface)

See also ?

coding 3-1

commands 3-1

definition 1-1

function 3-1

return codes 3-2

application program interface (API)

definition 1-1

AS/400 books I-1

AS/400 licensed program 1-4, 1-5

authorization I-1

automatic call 1-5

B

bibliography I-1

C

catchall record, definition 2-23

changing fields

in template records 2-14

changing host file cross-references 4-25

Class

ITEMVOID table 6-19

LAYMDSE table 6-16

MDSESALES table 6-15

SPECIALSALES Table 6-33

SPRSNSALES Table 6-32

VPLVOID table 6-20

Clear Retail File (CLRRTLF) command 4-10

COBOL/400

ADCS API coding example 3-1

data division specifications 2-24

data type 7 special handling 2-24

example 2-24, 3-3

ILE API coding example 3-6, B-1

source statements 1-2

command

examples

*STORE 2-2

add a keyed record 4-17

delete a file 4-15

delete a keyed record 4-17, 4-21

load a file 4-7

read a keyed record 4-17

receive a file 4-3

replace a keyed record 4-17

start a user program 4-19

types

ADCS 1-1

AS/400 CL 4-3

Host Command Processor 1-2

command diagrams

CLRRTLF (Clear Retail File) 4-11

CRTRTLF (Create Retail File) 4-13

CRTRTLSRC (Create Retail Source) 2-25

CVTFRMRTL (Convert From Retail File) 4-33

CVTTORTL (Convert To Retail File) 4-31

DLRRTL (Delete Retail File) 4-15

DLRRTLRC (Delete Retail Record) 4-22

ENDRTLPLLE (End Retail Polling Entry) 5-5

ENDRTLSVR (End Retail Server) 5-4

PRRTLRLPT (Print Retail Report) 7-2

PRRTLRLTPL (Print Retail Conversion Template) 2-26

RCVRTLF (Receive Retail File) 4-4

RMVRTLTNS (Remove Retail Transactions) 7-1

SBMRTLPG (Submit Retail Program) 4-20

SNDRTL (Send Retail File) 4-8

SNDRTLRC (Send Retail Record) 4-17

STRRTLPLLE (Start Retail Polling Entry) 5-4

STRRTLSVR (Start Retail Server) 5-4

WRKRTLTPL (Work with Retail Templates) 2-27

commands

ADCS 4-3

CL commands supported 4-3

Clear Retail File (CLRRTLF) 4-10

Convert From Retail File (CVTFRMRTL) 4-32

Convert To Retail File (CVTTORTL) 4-30

Create Retail File (CRTRTLF) 4-12

Delete Retail File (DLRRTL) 4-15

Delete Retail Record (DLRRTLRC) 4-21

LOAD (Load) 4-7

Receive Retail File (RCVRTLF) 4-3

Send Retail File (SNDRTL) 4-7

Send Retail Record (SNDRTLRC) 4-17

commands *(continued)*

- Submit Retail Program (SBMRTLPGM) 4-19
- Work with Retail Communications (WRKRTLCPM) 4-1

communications, SDLC 1-6

configuration

- nonswitched retail controllers B-1
- planning for 1-6
- switched line retail controllers B-2
- TCP/IP Host names B-2
- TCP/IP interfaces B-2
- X.25 retail controllers B-2

confirming deletion of host file cross-references 4-25

control language support

- retail file communications 4-3

conversion

- from retail files 4-28
- point-of-sale data field 2-23
- retail file considerations 2-22
- retail file description 1-2
- retail file requirements 2-23
- retail file system 1-2
- retail file templates 2-23

Convert From Retail File (CVTFMRRLF) 4-32

Convert To Retail File (CVTTORRLF) 4-30

CREA command

- HCP emulation 1-2
- I/O emulation 1-2

Create Retail Source (CRTRLSRC) command 2-25

cross-references, renaming host file 4-25

D

data conversion

- changing templates 5-1
- error log 4-35, 5-8
- routine 2-23

data description specifications (DDS)

- data type 7 special handling 2-24
- retail conversion template 2-23
- source statements 1-2

data field

- point-of-sale conversion 2-23

data type 7=Item code 2-24

data types

- valid combinations 2-18

DDS (data description specifications)

- data type 7 special handling 2-24
- retail conversion template 2-23
- source statements 1-2

delete a file

- on a retail controller 4-15

delete a keyed record

- in a retail controller file 4-21

Delete Retail File (DLTRRLF) command 4-15

Delete Retail Record (DLTRRLRCD) command 4-21

deleting a keyed record

- in a retail controller file 4-17

DeptNum

- ITEMVOID table 6-19
- LAYMDSE table 6-16
- MDSESALES table 6-15
- SPECIALSALES Table 6-33
- SPRSNSALES Table 6-32
- VPLVOID table 6-20

DISCOUNT Table 6-8

DISCREPANCY table 6-14

DiscType

- DISCOUNT Table 6-8
- ITEMDISCOUNT Table 6-17

displays

- Add Field in Template Record 2-15
- Add Host File Cross Reference 4-25
- Add Retail Polling Entry 5-2
- Add Retail Store Entry 2-4
- Add Retail Store Information 2-2
- Add Retail Template Record 2-12
- Change Field in Template Record 2-14
- Change Host File Cross Reference 4-25
- Change Retail Conversion Template 2-8
- Change Retail Store Information 2-3
- Change Retail Template Record 2-12
- Configure Retail Communications Application 2-1
- Confirm Delete of Host File Cross References 4-25
- Confirm Delete of Retail Conversion Templates 2-8
- Confirm Delete of Retail Polling Entries 5-2
- Confirm Delete of Retail Store Information 2-3
- Confirm Remove of Fields 2-15
- Confirm Remove of Template Records 2-13
- Convert from Retail File Format 4-28
- Convert to Retail File Format 4-29
- Copy Fields in Template Record 2-15
- Copy Records from Conversion Template 2-12
- Copy Retail Conversion Templates 2-8
- Create Retail Conversion Template 2-9
- Display Field in Template Record 2-15
- Display Retail Conversion Template 2-9
- Display Retail Polling Entry 5-1
- Display Retail Store Information 2-3
- Display Retail Template Record 2-13
- End Retail Polling Entry 5-5
- End Retail Server 5-4
- Exit Work with Retail Template 2-11
- Remove Retail Polling Entry 5-2
- Remove Retail Store Entry 2-4
- Rename Host File Cross Reference 4-25
- Rename Records in Retail Template 2-13
- Rename Retail Conversion Templates 2-9
- Rename Retail Store Information 2-4
- Retail Application Interface Suite 2-1, 4-1
- Retail Application Interface Suite for AS/400 4-27
- Select Retail Conversion Template 4-29

displays *(continued)*

- Start Retail Polling Entry 5-4
- Start Retail Server 5-4
- Work with Fields in Template Record 2-14
- Work with Host File Cross References 4-24
- Work with Records in Retail Template 2-11
- Work with Retail Communications 4-1, 4-2
- Work with Retail Conversion Templates 2-7
- Work with Retail File Conversion 4-27
- Work with Retail Polling Entries 5-1
- Work with Retail Store Information 2-2

distributed system node executive (DSNX)

- definition 1-4

DscVoid

- ITEMVOID table 6-19

DSNX (distributed system node executive)

- definition 1-4

dual controller support

DUMP command, HCP emulation 1-2

DUMP command, I/O emulation 1-1

E

EFT (Electronic Fund Transfer) 1-3

Electronic Fund Transfer (EFT) 1-3

emulation, ADCS

- adding a keyed record command 4-17
- deleting a keyed record command 4-21
- menus 4-1
- new host (new PSS) format files 4-3
- PC format files 4-2
- point-of-sale communications 4-1
- point-of-sale file types 4-2
- point-of-sale function 1-4
- print format files 4-2
- PSS format files 4-2
- reading a keyed record command 4-17
- replacing a keyed record command 4-17

emulation, HCP

- commands supported 4-24
- considerations 4-26
- conversion services 2-23
- definition 1-2
- message queues 4-26
- point-of-sale host system 1-4
- required steps 4-26
- retail user profile 4-26
- sense codes 4-27
- using 4-24

emulation, retail records 4-17

end-of-data (EOD) flag 3-2

environments

- See host system environments

escape messages

- Convert From Retail File (CVTFRMRTLFL) command 4-35

escape messages *(continued)*

- Convert to Retail File (CVTTORTLFL) command 4-32

examples

- ADCS emulation API, COBOL coding 3-1
- ADCS emulation API, RPG coding 3-1
- add retail store information 2-2
- adding or replacing a record in a retail controller file 4-19
- change field in template record 2-14
- change host file cross reference 4-25
- change retail conversion template 2-8
- change retail store information 2-3
- change retail template record 2-12
- configuring an X.25 retail network B-2
- configuring nonswitched retail controllers B-1
- configuring switched line retail controllers B-2
- creating a retail conversion template 2-20
- creating a retail database collection conversion template 2-21
- data type 7 special handling, COBOL coding 2-24
- data type 7 special handling, RPG coding 2-24
- delete field in template record 2-15
- delete host file cross reference 4-25
- delete retail conversion template 2-8
- delete retail polling entry 5-2
- delete retail store information 2-3
- delete retail template record 2-13
- deleting a record in a retail controller file 4-24
- display field in template record 2-15
- display retail conversion template 2-9
- display retail polling entry 5-1
- display retail store information 2-3
- host environment illustration 1-5
- in-store processor environment illustration 1-4
- Retail Connectivity Option server 2-5
- retail controller record update, COBOL coding 3-3
- TCP/IP Configuration B-2
- using HCP emulation 4-26

ExchRate

- TENDER table 6-9

F

FAIL-TYPE (FALTYP) values 3-2

FeeType

- ACCTFEE Table 6-25

field descriptions

- of retail conversion templates 2-10

files

- deleting 4-15
- keyed 2-7
- sequential 2-7
- structures 2-7

FROMMBR(*SEARCH) option 4-34

FuncType

- DISCREPANCY table 6-14
- MDSETOTALS Table 6-5

FuncType (continued)

TAXAMT Table 6-6
TAXCODE Table 6-7
TRANAMOUNT table 6-3

G

General Sales Application (GSA)

definition 4-19

Granting authorizations 1-1

GrossNeg

TRANAMOUNT table 6-3

GrossPos

TRANAMOUNT table 6-3

H

hardware requirements 1-6

HCP (host command processor)

commands 1-2
definition 1-2
emulation
 commands supported 4-24
 considerations 4-26
 conversion services 2-23
 definition 1-2
 message queues 4-26
 point-of-sale host system 1-4
 required steps 4-26
 retail user profile 4-26
 sense codes 4-27
 using 4-24

HEADER table 6-4

high-level source statements 2-24

host command processor (HCP)

commands 1-2
definition 1-2

host file cross references

adding 4-25
changing 4-25
confirming deletion 4-25
renaming 4-25
working with 4-24

host system environments

AS/400 system as an in-store processor 1-3
HCP 1-6
in-store processor 1-3
licensed program 1-5
point-of-sale network 1-5
retail file conversion 1-5
user polling 1-5

Hourly

ACCTDEP Table 6-24
ACCTFEE Table 6-25
ACCTNEW Table 6-22
ACCTPAY Table 6-23

Hourly (continued)

DISCOUNT Table 6-8
ISSUETNDR Table 6-21
ITEMDEPOSIT Table 6-18
ITEMDISCOUNT Table 6-17
ITEMVOID table 6-19
LAYCAN Table 6-26
LAYMDSE table 6-16
LAYRSTK Table 6-27
MDESALSALES table 6-15
MDSETOTALS Table 6-5
PAYINS table 6-30
PAYOUTS table 6-29
PICKUPS table 6-28
POSTVOID table 6-12
SPECIALSALES Table 6-33
SPRSNSALES Table 6-32
TAXAMT Table 6-6
TAXCODE Table 6-7
TENDER table 6-9
TENDERCHG table 6-10
TNDRCASH table 6-31
TRANSVOID table 6-11
VPLVOID table 6-20

I

in-store processor

environment 1-3
Retail Application Interface Suite 1-3

installation

commands 1-8
planning requirements 1-6

installing Retail Application Interface Suite

hardware requirements 1-6
software requirements 1-7

ISOCurNum

TENDER table 6-9

ISSUETNDR Table 6-21

ITEMDEPOSIT Table 6-18

ITEMDISCOUNT Table 6-17

ItemNum

DISCREPANCY table 6-14
ITEMDEPOSIT Table 6-18
ITEMDISCOUNT Table 6-17
ITEMVOID table 6-19
LAYMDSE table 6-16
MDESALSALES table 6-15
SPECIALSALES Table 6-33
SPRSNSALES Table 6-32
VPLVOID table 6-20

ITEMVOID table 6-19

K

KEYD command, I/O emulation 1-2

keyed files

- definition 1-1
- retail controller 2-7

keyed record

- adding 4-17
- deleting 4-17, 4-21
- reading 4-17
- replacing 4-17

L

LAYCAN Table 6-26

LAYMDSE table 6-16

LAYRSTK Table 6-27

load a file

- AS/400 to retail controller (SNDRTLFL command) 4-7
- Send Retail File (SNDRTLFL)
 - *STORE option 4-7

LOAD command

- description 4-7
- HCP emulation 1-2
- I/O emulation 1-1

LuxTax

- TAXAMT Table 6-6

M

manuals

- 3650 I-1
- 3680 I-1
- 4680 I-1
- 4684 I-1
- 4690 I-1
- AS/400 I-1
 - retail controller I-1
 - retail host I-1

MBROPT parameter

- RCVRTLFL command
 - reference matrix 4-6

MDSESALES table 6-15

MDSETOTALS Table 6-5

messages, escape

- Convert From Retail File (CVTFRMRTLFL)
 - command 4-35
- Convert to Retail File (CVTTORTLFL) command 4-32

N

NegVoided

- TRANSVOID table 6-11

NetCash

- TRANAMOUNT table 6-3

NetNonCash

- TRANAMOUNT table 6-3

NetView DM

- definition 1-4
- licensed program 1-4
- pass-through 1-3, 1-4

NonMerch

- LAYMDSE table 6-16
- MDSESALES table 6-15
- VPLIVOID table 6-20

nonswitched retail controllers

- configuration B-1

NTaxblDiscount

- MDSETOTALS Table 6-5

NTaxblFees

- MDSETOTALS Table 6-5

NTaxblRefund

- MDSETOTALS Table 6-5

NTaxblSales

- MDSETOTALS Table 6-5

NTxblamt

- VPLIVOID table 6-20

NTxbldisc

- VPLIVOID table 6-20

O

OperNum

- ACCTDEP Table 6-24
- ACCTFEE Table 6-25
- ACCTNEW Table 6-22
- ACCTPAY Table 6-23
- DISCOUNT Table 6-8
- DISCREPANCY table 6-14
- HEADER table 6-4
- ISSUETNDR Table 6-21
- ITEMDEPOSIT Table 6-18
- ITEMDISCOUNT Table 6-17
- ITEMVOID table 6-19
- LAYCAN Table 6-26
- LAYMDSE table 6-16
- LAYRSTK Table 6-27
- MDSESALES table 6-15
- MDSETOTALS Table 6-5
- PAYINS table 6-30
- PAYOUTS table 6-29
- PICKUPS table 6-28
- POSTVOID table 6-12
- SPECIALSALES Table 6-33
- TAXAMT Table 6-6
- TAXCODE Table 6-7
- TENDER table 6-9
- TENDERCHG table 6-10
- TNDR CASH table 6-31
- TRANAMOUNT table 6-3
- TRANSVOID table 6-11
- VPLIVOID table 6-20

options

- *SEARCH 4-34
- *STORE 2-2, 4-34

OrgOperNum

DISCREPANCY table 6-14

OrgTranNum

DISCREPANCY table 6-14

OrgWrkStaNum

DISCREPANCY table 6-14

P

pass-through

- ADCS 1-3, 1-4
- NetView DM 1-3, 1-4

PAYINS table 6-30

PAYOUTS table 6-29

PhyStorNum

- ACCTDEP Table 6-24
- ACCTFEE Table 6-25
- ACCTNEW Table 6-22
- ACCTPAY Table 6-23
- DISCOUNT Table 6-8
- DISCREPANCY table 6-14
- HEADER table 6-4
- ISSUETNDR Table 6-21
- ITEMDEPOSIT Table 6-18
- ITEMDISCOUNT Table 6-17
- ITEMVOID table 6-19
- LAYCAN Table 6-26
- LAYMDSE table 6-16
- LAYRSTK Table 6-27
- MDSESALES table 6-15
- MDSETOTALS Table 6-5
- PAYINS table 6-30
- PAYOUTS table 6-29
- PICKUPS table 6-28
- POSTVOID table 6-12
- SPECIALSALES Table 6-33
- SPRSNSALES Table 6-32
- TAXAMT Table 6-6
- TAXCODE Table 6-7
- TENDER table 6-9
- TENDERCHG table 6-10
- TNDRCASH table 6-31
- TRANAMOUNT table 6-3
- TRANSVOID table 6-11
- VPLIVOID table 6-20

PICKUPS table 6-28

planning

- installation 1-6
- requirements 1-6

point-of-sale

- definition xv

polling function

polling, user 1-5

POSTVOID table 6-12

PosVoided

TRANSVOID table 6-11

Print Retail Conversion Template (PRRTLPL) command 2-26

Print Retail Report (PRRTLPT) command 7-2
problem analysis A-1

procedures

point-of-sale 4-1

programmable store systems (PSS) 1-1

PRRTLPT (Print Retail Report) command 7-2

publications, list of I-1

PURG command

- HCP emulation 1-2
- I/O emulation 1-2

Q

QRAIS user profile authorizations I-1

QtySales

- LAYMDSE table 6-16
- MDSESALES table 6-15
- SPECIALSALES Table 6-33
- SPRSNSALES Table 6-32

QtyVoid

ITEMVOID table 6-19

Quantity

VPLIVOID table 6-20

R

RAIS

control language support 4-3

RAIS API (application program interface)

- qzcfpcmn service program 3-4
- using 3-4

RAIS Communications API

qzcfpcmn definition 1-1

randomizing routine D-1

RCMS (remote change management server)

definition 1-5

RCVRTLF (Receive Retail File) command 4-3

re-creating problems A-1

reading a keyed record

in a retail controller file 4-17

Reason

- DISCOUNT Table 6-8
- ITEMDISCOUNT Table 6-17
- LAYMDSE table 6-16
- MDSESALES table 6-15
- PAYINS table 6-30
- PAYOUTS table 6-29

receive file

retail controller to AS/400 (RCVRTLF command) 4-3

Receive Retail File (RCVRTLF)

- command 4-3
- command reference matrix 4-6
- MBROPT parameter
 - reference matrix 4-6

Receive Retail File (RCVRTLF) command 4-3 receiving

- retail files 4-3

RED command

- HCP emulation 1-2
- I/O emulation 1-2

reference matrix

- RCVRTLF command 4-6

Refunded

- TAXCODE Table 6-7

related printed information I-1

relational database support

- GSA relational database definition 1-2
- Retail Application Logical Interface for GSA 1-3

remote change management server (RCMS)

- definition 1-5

Remove Retail Transactions (RMVRTLTNS)

command 7-1

renaming host file cross-references 4-25

replacing a keyed record

- in a retail controller file 4-17

requirements

- before you use this book xv
- hardware 1-6
- planning requirements 1-6
- software
 - AS/400 system 1-7
 - minimum 1-7
 - retail controllers 1-7
 - System/370 host 1-7

Retail Application Interface Suite for AS/400

- functions provided 1-1
- introduction 1-1
- required knowledge xv
- who should use xv

Retail Application Logical Interface

- for GSA 1-3

retail communications

- AS/400 system as a host system 1-5
- AS/400 system as an in-store processor 1-3
- components
 - ADCS 1-4
 - DSNX 1-4
 - HCP 1-4
 - NetView 1-4
 - RAIS Application Programming Interface 1-4
 - RCMS (remote change management server) 1-5
 - retail file conversion system 1-4
 - SNA Upline Facility 1-4
- configuring for 2-1
- hardware requirements 1-6

retail communications (continued)

- planning for 1-6
- software requirements 1-7
- working with 2-4, 3-1, 4-1

Retail Connectivity Option

- ADXBFTBL (File Table Maintenance Program) 2-6
- ADXSRVTL (TCP/IP Server) 2-5
- in host environment 1-6
- in ISP environment 1-5
- point-of-sale file types 4-3

retail controller books I-1

retail controller configuration examples B-1

retail conversion

- changing templates 2-8
- confirming deletion of 2-8, 5-2
- copying templates 2-8
- create template 2-9
- display polling entry 5-1
- display template 2-9
- printing template 2-26
- record description templates 2-15
- rename template 2-9
- template example 2-20, 2-21
- template field descriptions 2-10
- working with templates 2-7

retail file communications

- description 4-1
- Retail Connectivity Option 4-1

retail file conversion

- displays 4-27
- system considerations 2-22

retail files

- clearing 4-10
- conversion 4-27
- conversion considerations 2-22
- conversion displays 4-27
- conversion system 1-2
- converting from 4-28, 4-32
- converting to 4-29
- deleting 4-15
- polling configuration 5-1, 5-5
- polling function 1-2
- receiving 4-3
- selecting conversion template 4-29
- sending 4-7

retail host books I-1

Retail Industry Program Support Services (RIPSS)

- definition 1-1
- HCP communications 1-7
- on a retail controller 1-1

retail programs

- submitting 4-19

retail records

- deleting 4-21
- emulation 4-17
- sending 4-17

retail source

creating 2-25

retail store information

adding 2-2
changing 2-3
confirm delete 2-3
display description 2-2
displaying 2-3
renaming 2-4
working with 2-2

retail template record

add retail file table entry 5-6
adding 2-12
adding polling entry 5-2
change retail file table 5-6
changing 2-12
confirm remove 2-13
copying 2-12
create retail file table 5-6
display file table 5-6
display retail file table entry 5-7
displaying 2-13
field descriptions 2-13, 5-2
File Table field descriptions 5-7
renaming 2-13
retail file table 5-5
retail file table entries 5-6
working with 2-11

retail template record fields

adding 2-15
changing 2-14
confirm remove 2-15
copying 2-15
descriptions 2-15
displaying 2-15
working with 2-14

retail templates

working with 2-26

return codes

API 3-2
examples 3-2

Returned

ACCTDEP Table 6-24
ACCTFEE Table 6-25
ACCTPAY Table 6-23
DISCOUNT Table 6-8
ITEMDEPOSIT Table 6-18
ITEMDISCOUNT Table 6-17
LAYMDSE table 6-16
MDSESALES table 6-15
SPRSNSALES Table 6-32
VPLIVOID table 6-20

RIPSS (Retail Industry Program Support Services)

definition 1-1
HCP communications 1-7
on a retail controller 1-1

RMVRTLTRN (Remove Retail Transactions)

command 7-1

RPG/400

ADCS API coding example 3-1
data type 7 special handling 2-24
example 2-24
ILE API coding example 3-6
input and output specifications 2-24
source statements 1-2

S**SalesPrsn**

SPRSNSALES Table 6-32

SaleType

SPECIALSALES Table 6-33

Save command 1-9**SDLC communications 1-6****Send Retail File (SNDRTLF) command 4-7****Send Retail Record (SNDRTLRCR) command 4-17****SeqNum**

ITEMDISCOUNT Table 6-17

sequential files

definition 1-1
retail controller 2-7

service information

problem analysis A-1

SNA upline facility (SNUF)

configuration 1-6
definition 1-3

SNUF (SNA upline facility)

configuration 1-6
definition 1-3

SNUF device

configuration
for ADCS 4-27
for point-of-sale pass-through 4-27

software requirements 1-7**source statements**

COBOL/400 1-2
DDS 1-2
high level 2-24
RPG/400 1-2

special program considerations

Chain Sales Application 2-20
General Sales Application 2-19
Supermarket Sales Application 2-19

SPECIALSALES Table 6-33**SPRSNSALES Table 6-32****start a user program**

retail controller 4-19

store systems 1-1**submit a retail job**

retail controller 4-19

Submit Retail Program (SBMRTLPGM) command 4-19

SUP command

HCP emulation 1-2
I/O emulation 1-2

switched line retail controllers

configuration B-2

T

Table Names

ACCTDEP Table 6-24
ACCTFEE Table 6-25
ACCTNEW Table 6-22
ACCTPAY Table 6-23
DISCOUNT Table 6-8
DISCREPANCY table 6-14
HEADER table 6-4
ISSUETNDR Table 6-21
ITEMDEPOSIT Table 6-18
ITEMDISCOUNT Table 6-17
ITEMVOID table 6-19
LAYCAN Table 6-26
LAYMDSE table 6-16
LAYRSTK Table 6-27
MDESSESALES table 6-15
MDSETOTALS Table 6-5
PAYINS table 6-30
PAYOUTS table 6-29
PICKUPS table 6-28
POSTVOID table 6-12
SPECIALSALES Table 6-33
SPRSNSALES Table 6-32
TAXAMT Table 6-6
TAXCODE Table 6-7
TENDER table 6-9
TENDERCHG table 6-10
TNDRCASH table 6-31
TRANAMOUNT table 6-3
TRANSVOID table 6-11
VPLIVOID table 6-20

TAX

TAXAMT Table 6-6
TAXCODE Table 6-7

TAXAMT Table 6-6

TaxblDiscount

MDSETOTALS Table 6-5

TaxblFees

MDSETOTALS Table 6-5

TaxblRefund

MDSETOTALS Table 6-5

TaxblSales

MDSETOTALS Table 6-5

TaxCode

TAXCODE Table 6-7

TAXCODE Table 6-7

TaxTbl

TAXCODE Table 6-7

TaxType

TAXAMT Table 6-6
TAXCODE Table 6-7

TCP/IP interfaces

configuration B-2

TDate

ACCTDEP Table 6-24
ACCTFEE Table 6-25
ACCTNEW Table 6-22
ACCTPAY Table 6-23
DISCOUNT Table 6-8
DISCREPANCY table 6-14
HEADER table 6-4
ISSUETNDR Table 6-21
ITEMDEPOSIT Table 6-18
ITEMDISCOUNT Table 6-17
ITEMVOID table 6-19
LAYCAN Table 6-26
LAYMDSE table 6-16
LAYRSTK Table 6-27
MDESSESALES table 6-15
MDSETOTALS Table 6-5
PAYINS table 6-30
PAYOUTS table 6-29
PICKUPS table 6-28
POSTVOID table 6-12
SPECIALSALES Table 6-33
SPRSNSALES Table 6-32
TAXAMT Table 6-6
TAXCODE Table 6-7
TENDER table 6-9
TENDERCHG table 6-10
TNDRCASH table 6-31
TRANAMOUNT table 6-3
TRANSVOID table 6-11
VPLIVOID table 6-20

template

definition 1-2
GSA relational database template 1-2
polling function 1-2

TENDER table 6-9

TENDERCHG table 6-10

TNDRCASH table 6-31

TndrType

ISSUETNDR Table 6-21
PAYINS table 6-30
PAYOUTS table 6-29
PICKUPS table 6-28
TENDER table 6-9
TENDERCHG table 6-10
TNDRCASH table 6-31

TOMBR(*STORE) option 4-35

TRANAMOUNT table 6-3

TranNum

ACCTDEP Table 6-24
ACCTFEE Table 6-25

TranNum *(continued)*

ACCTNEW Table 6-22
 ACCTPAY Table 6-23
 DISCOUNT Table 6-8
 DISCREPANCY table 6-14
 HEADER table 6-4
 ISSUETNDR Table 6-21
 ITEMDEPOSIT Table 6-18
 ITEMDISCOUNT Table 6-17
 ITEMVOID table 6-19
 LAYCAN Table 6-26
 LAYMDSE table 6-16
 LAYRSTK Table 6-27
 MDSESALES table 6-15
 MDSETOTALS Table 6-5
 PAYINS table 6-30
 PAYOUTS table 6-29
 PICKUPS table 6-28
 POSTVOID table 6-12
 SPECIALSALES Table 6-33
 SPRSNSALES Table 6-32
 TAXAMT Table 6-6
 TAXCODE Table 6-7
 TENDER table 6-9
 TENDERCHG table 6-10
 TNDRCASH table 6-31
 TRANAMOUNT table 6-3
 TRANSVOID table 6-11
 VPLIVOID table 6-20

TRANSVOID table 6-11**trickle transmit, definition C-1****TTime**

ACCTDEP Table 6-24
 ACCTFEE Table 6-25
 ACCTNEW Table 6-22
 ACCTPAY Table 6-23
 DISCOUNT Table 6-8
 DISCREPANCY table 6-14
 HEADER table 6-4
 ISSUETNDR Table 6-21
 ITEMDEPOSIT Table 6-18
 ITEMDISCOUNT Table 6-17
 ITEMVOID table 6-19
 LAYCAN Table 6-26
 LAYMDSE table 6-16
 LAYRSTK Table 6-27
 MDSESALES table 6-15
 MDSETOTALS Table 6-5
 PAYINS table 6-30
 PAYOUTS table 6-29
 PICKUPS table 6-28
 POSTVOID table 6-12
 SPECIALSALES Table 6-33
 SPRSNSALES Table 6-32
 TAXAMT Table 6-6
 TAXCODE Table 6-7

TTime *(continued)*

TENDER table 6-9
 TENDERCHG table 6-10
 TNDRCASH table 6-31
 TRANAMOUNT table 6-3
 TRANSVOID table 6-11
 VPLIVOID table 6-20

Txblamt

VPLIVOID table 6-20

Txbldisc

VPLIVOID table 6-20

U**user polling 1-5****user profile authorizations I-1****utility trailer record C-1****V****VatTax**

TAXAMT Table 6-6

VPLIVOID table 6-20**W****Weightltm**

MDSESALES table 6-15

Work with Retail Communications (WRKRTLCLMN)**command 4-1****Work with Retail Templates (WRKRTLPL)****command 2-26****working with host file cross-references 4-24****WrkStaNum**

ACCTDEP Table 6-24
 ACCTFEE Table 6-25
 ACCTNEW Table 6-22
 ACCTPAY Table 6-23
 DISCOUNT Table 6-8
 DISCREPANCY table 6-14
 HEADER table 6-4
 ISSUETNDR Table 6-21
 ITEMDEPOSIT Table 6-18
 ITEMDISCOUNT Table 6-17
 ITEMVOID table 6-19
 LAYCAN Table 6-26
 LAYMDSE table 6-16
 LAYRSTK Table 6-27
 MDSESALES table 6-15
 MDSETOTALS Table 6-5
 PAYINS table 6-30
 PAYOUTS table 6-29
 PICKUPS table 6-28
 POSTVOID table 6-12
 SPECIALSALES Table 6-33
 SPRSNSALES Table 6-32
 TAXAMT Table 6-6

WrkStaNum *(continued)*

TAXCODE Table 6-7
TENDER table 6-9
TENDERCHG table 6-10
TNDRCASH table 6-31
TRANAMOUNT table 6-3
TRANSVOID table 6-11
VPLIVOID table 6-20

X

X.25 retail controllers

configuration example B-2

Z

ZERO command

HCP emulation 1-2
I/O emulation 1-2