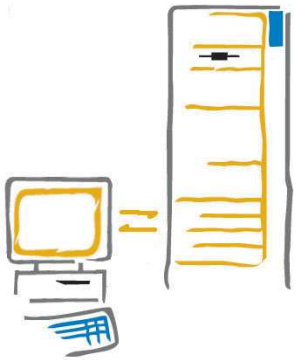
ETU/36®

File Transfer Utility
For the System/36 and Advanced/36



Reference Guide

RINGDALE®
Connecting people and information

NLynx

SOFTWARE LICENSE AGREEMENT

NOTICE
=======
YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE OPENING THIS PACKAGE. OPENING THIS PACKAGE INDICATES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THEM, YOU SHOULD PROMPTLY RETURN THE PACKAGE UNOPENED, AND YOUR MONEY WILL BE RETURNED. Ringdale provides the computer software program contained on the medium in the package (The "Software") and licenses its use. You assume responsibility for the selection of the Software to achieve your intended results, and for the installation, use and results obtained from the Software.

DEFINITIONS
============
The following definitions apply to the terms as they appear in this Agreement:

- "Ringdale" means Ringdale, Inc. Ringdale is a division of Network Technology Plc. Group of Companies.
- NLynx or NLynx Technologies is the product name of Ringdale, Inc.
- "You" and "Your" refer to any person or entity that acquires or uses this Package.
- "Software" means the computer programs contained in this Package, together with all codes, techniques, software tools, formats, designs, concepts, methods, and ideas associated with the program.
- "Package" means the Software, hardware, manual(s), and other items accompanying this Agreement.

PERMITTED AND INTENDED USES
============================
Ringdale grants You a non-exclusive license to use the Software in this Package according to the terms set forth below. You may:

* Operate the Software on one computer at a time.

* Make one backup copy of the Software, which automatically becomes the property of Ringdale and is subject to this Agreement.

* Transfer the Software to a single hard disk drive or other permanent storage device for Your own use and keep the original Software as a second backup copy.

* Make such other use of the Software as is specifically permitted by the Installation Manual relating to the Software.

PROHIBITED USES
==========
You may not:
* Make the Software available to any person or entity other than Your employees, who must use the Software as specified above. You may, however, after written notification to Ringdale, transfer all (but not a lesser portion) of the Software and related manuals to another person or entity, which in turn will be subject to this Agreement.
* Modify the Software or merge it with another program. Any modified or merged portion of the Software is subject to this Agreement.
* Reverse-engineer, disassemble, decompile, or attempt to discover the source code to the Software.
* Translate or create derivative works based on the Software.
* Remove, obscure, or alter any notice of the Ringdale copyright or other proprietary rights related to the Software.
* Sub-license, sell, lend, rent, or lease the Software or any portion of the Software, or use it for any purpose other than Your internal purposes.
* Copy any portion of the Software, except to make a backup copy or to transfer to a hard disk drive or other permanent storage device.
* Transfer the Software or any direct product to any person or entity in violation of the United States Export Administration Act.

The Software involves valuable proprietary rights of Ringdale, Inc. and others. Ringdale and others retain title to and ownership of the Software and all copyright, trade secret, trade name, trademark, and other property rights related to the Software, regardless of the form in which the original or other copies exist. You may not violate these rights, and You must take appropriate steps to protect Ringdale's rights. Ringdale may at any time replace, modify, alter, improve, enhance, or change the Software.

Both the License and Your right to use the Software terminate automatically if You violate any part of this Agreement. In the event of termination, You must immediately destroy all copies of the Software or return them to Ringdale.

LIMITED WARRANTY
==========
Ringdale warrants that, upon delivery by Ringdale, the diskettes or other media on which the Software is distributed will be free from defects in materials for a period of 90 days. The Software itself IS LICENSED TO YOU ON AN "AS IS" BASIS. If the diskette or other media fails to comply with the warranty set forth above, Ringdale will replace the diskette or other media. You must, however, return all copies of the Software; along with a copy of Your paid invoice, to Ringdale within 90 days of the date You receive the Software.

This warranty gives You specific legal rights; in some jurisdictions You may have other rights under certain statutes which imply non-excludable warranties.

Except as set out above and to the maximum extent permitted by law:

(A) RINGDALE MAKES NO REPRESENTATIONS, WARRANTIES, OR CONDITIONS WHETHER EXPRESS OR IMPLIED, BY STATUTE OR OTHERWISE, INCLUDING BUT NOT

LIMITED TO IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE;

(B) RINGDALE WILL NOT BE LIABLE FOR SPECIAL, INCIDENTAL, CONSEQUENTIAL, INDIRECT OR OTHER SIMILAR DAMAGES EVEN IF RINGDALE OR ITS AGENT HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS MEANS RINGDALE IS NOT RESPONSIBLE OR LIABLE FOR DAMAGES OR COSTS INCURRED BECAUSE OF LOSS OF TIME, LOSS OF DATA, LOSS OF PROFITS OR REVENUE, OR LOSS OF USE OF THE SOFTWARE. IN ADDITION, RINGDALE IS NOT RESPONSIBLE OR LIABLE FOR DAMAGES OR COSTS INCURRED IN CONNECTION WITH OBTAINING SUBSTITUTE SOFTWARE, CLAIMS BY OTHERS, INCONVENIENCE OR SIMILAR COSTS;

(C) RINGDALE IS NOT LIABLE FOR ANY DAMAGES TO YOU OR ANY OTHER PERSON IN EXCESS OF THE PRICE YOU PAID FOR THE LICENSE TO USE THE SOFTWARE, REGARDLESS OF THE FORM OF THE CLAIM;

(D) ADDITIONAL STATEMENTS BY AGENTS, EMPLOYEES, DISTRIBUTORS AND DEALERS OF RINGDALE, SUCH AS DEALER ADVERTISING OR PRESENTATIONS, DO NOT CONSTITUTE WARRANTIES OR CONDITIONS BY RINGDALE, DO NOT BIND RINGDALE AND SHOULD NOT BE RELIED UPON.

This warranty does not cover claims that the disk or material has been lost, stolen, or damaged by accident, misuse or unauthorized modification.

Though Ringdale is not responsible for maintaining or helping you use the Software, Ringdale does at its discretion offer support and service programs for certain software products. To receive the benefits of these programs, You must complete and return the enclosed User Registration Card.

This Agreement constitutes the entire agreement and supersedes any prior agreement between Ringdale and You concerning the contents of this Package. Ringdale is not bound by any provision of any purchase order, receipt, acceptance, confirmation, correspondence, or otherwise, unless Ringdale specifically agrees to the provision in writing. Any written or oral information or advice given by Ringdale dealers, distributors, agents, or employees will in no way increase the scope of this Agreement, nor may You rely on any oral or written communications from any such persons. This Agreement can be amended, modified, or waived only if the change is written and signed by an authorized officer of Ringdale.

Legal Notices

This document is for use with the Ringdale, Inc. ETU/36 product.

June 2006 Edition

This edition applies to Version 4.108 of ETU/36 and to all subsequent versions and modifications until otherwise indicated in new editions or updates to this publication.

It is possible that this publication could contain technical inaccuracies and typographical errors.

Address comments to: Ringdale Inc.
Technical Support
101 Halmar Cove
Georgetown, Texas 78628
(512) 288-9080
www.ringdale.com
support@ringdale.com

Preface

This manual is intended for users of the ETU/36 Emulator Transfer Utility and assumes the reader has a basic working knowledge of the IBM System/36 or the IBM Advanced/36, SSP, the PC Operating System and the emulation software that is running on the PC.

Trademark Recognition

Throughout this manual, the terms IBM, IBM PC, IBM PC/XT, IBM PC/AT, IBM PS/2 and IBM Enhanced 5250 Emulator are used in explanations and instructions. These terms are trademarks of International Business Machines Corporation.

3Xtwin, ES32, ES/95, ETU/36, and ETU/400 are trademarks of Ringdale, Inc.

DIF is a trademark of Software Arts, Inc.

Lotus 1-2-3 is a trademark of Lotus Development Corporation.

Mac is a trademark of Apple Corporation.

Microsoft® is a registered trademark and Windows® is a registered trademark of Microsoft Corporation.

IBM® is a registered trademark and AS/400® is a registered trademark of International Business Machines Corporation.

Table of Contents

Chapter 1 1

INTRODUCTION 1
Product Description 1-1
Prerequisite Programs 1-2
Restrictions 1-2
Considerations 1-3

Chapter 2 1
INSTALLATION 2-1

Chapter 3 1
OPERATION: ETU/36 MENU 3-1

Chapter 4 1
FILE TRANSLATION FACILITY 4-1

Appendix A 1
ASCII/EBCDIC PRINTABLE CHARACTERS A-1

Appendix B 1
International Translation Tables B-1

Appendix C 1
Messages C-1

Appendix D 1
ETU/36 Problem Guide D-1

Appendix E 1
ETU/36 Automation E-1

Appendix F 1
CREATING F & I SPECIFICATIONS F-1

Appendix G 1
GLOSSARY G-1

Appendix H 1
INDEX H-1

Chapter 1
INTRODUCTION

Product Description

The System/36 Emulator Transfer Utility (ETU) allows the interchange of data files between the IBM System/36 and a PC or Mac emulating an IBM 5251 display station via ES32 emulation software. By using ETU, you can perform the following functions from your workstation:

- Transfer data between disk or diskette files n the workstation and files on the System/36.
- Translate files to and from ASCII Text, BASIC Sequential, DIF and Tab formats.
- Allocate files on the workstation.
- Rename and delete files on the workstation.
- Direct ETU to a specific workstation subdirectory with file name path support.
- Store workstation files on the host without translation (binary file transfer: *SAVE).
- Specify a System/36 file3 as a save file (in *SAFV format).
- Test for the existence of files on the workstation.
- Translate files for format and data compatibility between the workstation and the System/36.
- Work with data files, library members and print files.

ETU resides on the System/36 and interfaces with the workstation emulation software. The utility can be executed from the ETU/36 menu, from an OCL procedure calling an ETU/36 function, or from the command entry display with a direct procedure call.

The ETU translation facility supports the translation of data with user-alterable translation tables. Both character-oriented and field-oriented translation capabilities are supported. Translation table source is stored in source library members and can be maintained with the System/36 Source Entry Utility (SEU).

A translation table compiler is provided to edit and convert character-oriented tables into files for use in text translation. Field-oriented translation tables are used for DIF and Basic Sequential formats, and are stored as object library members. All translation functions can be submitted to the job queue to better control the workload that remote users present to the System/36.

Prerequisite Programs

- ✓ System/36 SSP release 3 or higher and SEU.
- ✓ IBM Windows (98, 2000, XP or higher) based PC running the ES32 5250 emulation software (or other ETU compatible emulation – ask your emulation provider if they support ETU).

  or

  Apple Mac running the NLynx TCP/Axcess or MacMidrange Client.

Restrictions

Record lengths from one to 4096 bytes are supported.

The interactive functions of ETU (ALLOCATE, RENAME, DELETE, TESTFILE, FILETOPC, LIBRTOPC, PRNTTOPC, FILEFRPC, LIBRFRPC, and PRNTFRPC) must be initiated from your workstation.

Mac print data cannot be printed on the System/36 host.

ETU can be interrupted by the use of the Attention key and another job can be initiated provided the additional job is not an interactive ETU function.

System/36 Folders are not supported under this version of ETU.

Considerations

Security

Any file on the System/36 with a supported file type may be transferred with ETU. This may allow users to copy sensitive data. If you have security considerations, you should restrict the use of ETU by using the security provided with the System/36 SSP.

Local Data Area

The ETU function TESTFILE optionally uses 12 bytes of the user local data area (LDA) to store the results of the test. The user can specify the offset. This data can be used to condition later job steps.

If any of the TOPC (FILETOPC, LIBRTOPC AND PRNTTOPC) functions of ETU are interrupted during its operation, the file on the workstation may be left open. This can occur if you press the Attention key and cancel the job from the Inquiry Menu, or if the session is abnormally terminated by a communications problem. Executing any other ETU diskette function, such as TESTFILE, will close the file. If executing any of the TOPC functions and transferring files to or from a diskette on your workstation, do not remove the diskette from the drive until the function has finished executing completely.

Compatibility

This release of ETU supports all functions and procedure calls of previous releases of ETU. It is compatible with all 5250-emulation software developed by NLynx Technologies.

3XTwin	version 2.3 and higher
ES/3XTwin	all
ES/Remote	all
ES/PCI	all
ES/TCP	all
TCP/Axcess	all
MacMidrange Client	all

ETU/36 remains compatible with legacy emulation programs from Emerald Technology, KMW, Local Data, Andrew Network Products, Decision Data, NLynx Systems, and NLynx Technologies. DOS based emulation programs may be supported by using ETU's IBMTRAN (see Chapter 2 -1). Emulation programs from other vendors may provide support for ETU file transfer – ask your emulation providers to be sure.

ETU Compatibility for DOS Based Emulation

ETU works with all ES32 emulation products.

For non-ES32 DOS-based products, there is a compatibility disk with a program called IBMTRAN that enables any DOS-based emulation to work. This software only works with DOS software using serial COM ports. IBMTRAN was never ported to the Windows 32-bit environment; therefore, IBMTRAN doesn't work with any of the Windows 32-bit Operating Systems. A PC with Windows 3.1 could use it. It will not work with Windows95, 98, NT, 2000 or XP. IBMTRAN is no longer supported.

If you would like to try it, without support, you can get it from our website at http://www.nlynx.com/html/tb-etu36.htm click on the link:

ETU Compatibility Program for REAL DOS Only.

The compatibility program is designed to work through the Application Program Interface (API) of the emulation product to provide all the DOS functions required to complete the ETU tasks. If you are using a DOS based non-ES32 emulation, you must copy all of the programs included in the IBMTRAN file into the same directory where the emulation program resides.

Once the program have been copied, you must follow these instructions each time you want to use ETU with the non-ES32 DOS based emulator:

1. Load the emulator and establish a session with the host.
2. Hot key over to DOS and enter IBMTRAN. When the program is loaded it will automatically hot-key back to emulation.

Any ETU function can now be executed normally.

The IBMTRAN program remains active until it is stopped by hot-keying to DOS and pressing ALT-Break (for use with batch jobs, you may enter “IBMTRAN X” to unload IBMTRAN when control returns to DOS).

Note: if your program has a separate API program, make sure that the API program is running or your program will not be compatible with ETU.

Chapter 2
INSTALLATION

Inventory Checklist

ETU/36 is distributed on a diskette (8 or 5¼ inch) for the System/36 or a ¼ inch cassette for the Advanced System/36. The distribution media contains the following files:

PCTRAN	Library files; contains all System/36 programs and procedures.
PCTX.E@A	Compiled EBCDIC-to-ASCII variable record length translate file for Windows 32-bit PCs.
PCTXE@AF	Compiled EBCDIC-to-ASCII fixed record length translate file for Windows 32-bit PCs.
PCTX.A@E	Compiled ASCII-to-EBCDIC variable record length translate file for Windows 32-bit PCs.
PCPR.A@E	Print translate file for PC-DOS for Windows 32-bit PCs.
PCMC.E@A	Compiled EBCDIC-to-ASCII variable length translate file for Apple Mac.
PCMC.A@E	Compiled ASCII-to-EBCDIC variable length translate file for Apple Mac.
PCMCE@AF	Compiled EBCDIC-to-ASCII fixed length translate file for Apple Mac.

IBMTRAN

Earlier shipments of ETU/36 included a 3½-inch diskette containing the program IBMTRAN. This program is the ETU Compatibility Program for DOS. See section "**ETU Compatibility for DOS Based Emulation** " in Chapter 1. This program will only work on *native* MS-DOS or equivalent operating system. It *will not* work in the MS-DOS window under Windows (any version). For this reason, the compatibility diskette is no longer included in the product packaging. To obtain a copy, please download it from:

ftp://ftp.nlynx.com/Support/Nlynx/Midrange/EmeraldSeries/ETU/IBMTRAN/

Installation Commands

If you are upgrading an existing version of ETU, back up your system library before installing the new ETU version.

NOTE: If System/36 system security is active, you must be signed on as a security officer in order to install ETU/36 software.

If you are installing from *diskette*, enter the following command:

JOBSTR IETU,IETU,NOSAVE,#LIBRARY

If you are installing from a *tape cartridge* (cassette), enter the following command:

JOBSTR IETU,IETU,NOSAVE,#LIBRARY,,,,TC

These messages will be displayed:

JOBSTR procedure is running.
BLDLIBR procedure is running.
SYS-2594 ETUXFER Trying To Copy Privileged Module...

You are placed at the command line where you began. Take *Option 0* to respond to the last message, if it displays. Access the host machine serial number with the OWNERID command; PCTRAN must be your current session library. If necessary, press Print Screen.

NOTE: Make sure that at least 300 blocks are available on disk for the PCTRAN library.

Host Registration

ETU is licensed for single host system. After installation, you will need to obtain a host registration number from Ringdale, Inc. (see the enclosed Host Registration sheet for instructions).

When you have obtained your host registration number, start the installation procedure by entering **OWNERID** from your ETU library. A prompt will appear:

REGISTRATION CODE: XXXXXXXXXXXXX

Type your host registration number over the X's and then press the ENTER key or F7 to exit the OWNERID screen.

ETU can now be run indefinitely on the registered host system. If you reinstall ETU or its accompanying data files, you may need to reinstall the host registration number.

The Character Separator Option

You can also use the OWNERID command to change the decimal separator character from a period (.) to a comma (,). Choose the separator character you want at the prompt that displays when you enter the OWNERID command. The separator character default is a period.

International Translation Tables

For international users: To install foreign language tables, see the "International Translation Tables" notice included with this release.

Chapter 3

OPERATION: ETU/36 MENU

To use the Emulator Transfer Utility (ETU), you must first establish a session with System/36 from your workstation. A session is established by simply signing onto the System/36. Consult your system operator for instructions on passwords, and the library name where the ETU programs can be accessed. Contained in the ETU is the menu ETU36, which provides you with a convenient starting point. Once you have successfully signed on, you may execute ETU procedures in any of the following ways:

- Selecting a custom user menu item or a custom user procedure that calls an ETU procedure.
- Entering the complete ETU/36 procedure call directly on the command line.
- Prompting an ETU procedure by selecting a menu item from the ETU/36 menu (first enter MENU ETU36 on the command line).
- Prompting an ETU procedure by entering "ETU/36 procedure" without any of the associated parameters, where procedure is the ETU procedure (such as ALLOCATE, FILETOPC, LIBRFRPC) to be executed. For example: ETU36 ALLOCATE

The ETU procedures that require communication with the workstation (ALLOCATE, DELETE, TESTFILE, RENAME, FILETOPC, LIBRTOPC, PRNTTOPC, FILEFRPC, LIBRFRPC, PRNTFRPC) can only be executed on a workstation emulating a 5250 workstation with the appropriate workstation software. These interactive jobs must be run from the workstation that is to have the disk/diskette altered, and cannot be running in the inquiry mode. ETU procedures that do not communicate with the PC (XLT36FIL, XLT36PRT, XLTPCFIL, XLTPCPRT, EDITABLE, COMPILE) can be run from any workstation, submitted to the job queue (with JOBQJ, or released from the workstation (with EVOKE).

Data translation (reformatting) is performed on the S/36. For example, when a file is being transferred to the workstation and translated into a workstation format, the translation takes place before the data transfer occurs. Similarly, if a file is being transferred from the workstation and translated from a workstation format, the translation takes place after the file has been completely moved to the S/36 disk. It is possible to perform the data translation in batch (released from the workstation) before transferring a data file to the workstation, by using the

translation-only procedures provided in the ETU. This will allow you to continue to work with the display while the translation takes place.

The release level of the ETU program can be determined by executing the ETU procedure ETUVER. The ETU release level is also displayed on the screen whenever a transfer is done between the S/36 and the workstation.

Workstation File Name

IMPORTANT NOTE: The ETU/36 menu and its accompanying function calls will only work with certain versions of emulation software. See "Operation" in Chapter 2 for additional information.

The ETU/36 menu and its accompanying set of function calls is designed to work with several types of microcomputers, including “IBM-compatible” PCs running ES32, and the Apple Mac running TCP/Axcess. When requesting a file name for a microcomputer attached to the S/36, the term "workstation file name" is used. A workstation file name can contain up to 80 characters. Do **not** use quotes around the workstation file name.

Because of the various microcomputers supported by the ETU, the workstation file name must follow the format required by the operating system of the attached microcomputer. Listed below are brief descriptions of the filename formats for IBM DOS and the Apple Mac. Refer to your operating system guides for additional information.

IBM PC DOS
Format: drive: \ pathname \ workstation_file_name.ext

Example: C:\SUBDIR1\EXAMPLE.TXT
where:
drive is C:
pathname is SUBDIR1
workstation_file_name is EXAMPLE.TXT

The filename may be up to 8 characters long; an extension (.ext) is optional and may be up to 3characters long.

Apple Mac
Format: disk:pathname:workstation_file_name

Example: APPLEDRIVE:FOLDER1:WORKSHEET

where:
disk is APPLEDRIVE
pathname is FOLDER 1

workstation_file_name is WORKSHEET

Mac Users: Important Notes

Most of the information in this chapter is relevant to both PC and Mac users. However, some information is specific only to the Mac user. Such information Is expressed in the following format:

Mac Users: This is how Mac-specific information is presented.

The information that is specific to the Mac user has to do with differences between Mac and PC interactions with ETU. For instance, the commands PRNITOPC and PRNTFP.PC, used to transfer and translate PC-DOS print files from the PC to the System/36, serve no function for the Mac and should not be used.

There are two other differences between Mac and PC interactions with ETU. First, for ETU/36 options requiring a translation table file name value, Mac users must change the preset default. Second, for some ETU/36 options, Mac users have access to standard Mac directory dialog boxes. These issues are discussed below.

Specifying a Translation Table File Name

Many of the ETU procedures include the option to translate data. If you choose to translate data, then you must change the preset default translation table, which exists for translating PC data, not Mac data.

For example, Screen 3-1 is displayed when you select ETU/36 option #13, used to transfer data from the host to the workstation. The "Reformat type" parameter is set to *NONE indicating no translation. If you specify *TEXT, Screen 3-2 is displayed. Note that the parameter "Translate table file" is set to PCTX.E@A, used to translate the EBCDIC-to-PC-ASCII variable-length records. To translate the EBCDIC-to-Mac-ASCII variable-length records, you must specify the table PCMC.E@A.

Screen 3-1

```
      ETU36 PROCEDURE
      Personal Computer Data Transfer and File Utility

      TRANSFER a file to the PC ............................. FILETOPC

      S/36 file name .........................................
      S/36 file date .........................................    (O)

      Reformat type: *DIF, *BASICS, *TAB, *TEXT, *SAVE, *NONE     *NONE  (O)









      Cmd 3 - Previous Menu      Cmd 7 - Cancel
```

Screen 3-2

```
                              ETU36 PROCEDURE
                  Personal Computer Data Transfer and File Utility

    TRANSFER a file to the PC ............................. FILETOPC

    S/36 file name ........................................ COL198

    S/36 file date .........................................              (O)

    Reformat type: *DIF, *BASICS, *TAB, *TEXT, *NONE........ *TEXT        (O)

    Translate table file .................................. ETOA1         (O)

    Number of records for the translation work file ........ 1000          (O)

    Record length for the translation work file ............ 128           (O)









 Cmd 2 - Page Back                                    Cmd 7 - Cancel
```

Table 3-1 lists the ETU procedures that can be used either to translate data or to edit or compile translation files. It also lists the procedures' corresponding ETU/36 menu numbers and translate tables.

Table 3-1

ETU Command	ETU/36 Menu Option	Translate Table		
		Character-Oriented	Field-Oriented	Source File
XLT36FIL	8	PCMC.E@A or PCMCEOAF	E@AM	
XLTPCFIL	10	PCMC.A@E	A@EM	
FILETOPC	13	PCMC.E@A or PCMCEOAF	E@AM	
L!BERTOPC	14	PCMC.E@A or PCMCE@AF	E@AM	
FiLEFRPC	16	PCMC.A@E	A@EM	
LIBRFRPC	17	PCMC.A@E	A@EM	
EDITABLE	20			E@AM, E@AFM, or A@EM
COMPILE	21			E@AM, E@AFM, or A@EM

Using Mac Directory Dialog Boxes

For Mac users, ETU provides directory dialog boxes for some of the ETU/36 menu options. You can use a directory dialog box instead of entering a value into a menu's workstation file name prompt. For example, the menu in Screen 3-3 is displayed when you select ETU/36 menu option #16, used to transfer a file from the workstation to the host. You can specify with the workstation file name prompt the location of the Mac file, or you can press CMD 11 to display a Mac directory dialog box.

NOTE: For the DELETE and RENAME procedures, the directory dialog boxes are unavailable.

```
                              ETU36 PROCEDURE
                Personal Computer Data Transfer and File Utility


Transfer PC file to the S/36 ............................... FILEFRPC

Workstation File Name :


S/36 file name ..........................................

S/36 file record length..................................... 128            (O)

Reformat type :*DIF, *BASICS, *TAB, *TEXT, *SAVE, *NONE ..... *NONE          (O)









       Cmd 3 - Previous Menu      Cmd 7 - Cancel      Cmd 11 - MAC Prompt
```

You use an ETU directory dialog box as you would use any standard Mac directory dialog box.

For information about using directory dialog boxes, refer to your Mac System Software User's Guide.

NOTE: If you cancel from a dialog box, you will receive an ETU user message with three options. Select option #3 to return to the ETU/36 menu.

ETU36 Menu

The ETU36 menu supports all ETU procedures. The menu displays when you issue the command **MENU ETU36**. Select a menu option, and you will be prompted for the parameters required by the procedure you selected.

Screen 3-5

```
Menu - ETU36                                                Workstation ID - W8
                        Emulator Transfer Utility S/36
                    COPYRIGHT (c) 1985, Emerald Technology Inc.

              PC Functions                          Transfer Functions
 1. Allocate New PC File                13. S/36 File         -> PC File
 2. Rename PC File                      14. S/36 Libr Member  -> PC File
 3. Delete PC File                      15. S/36 Print Item   -> PC Print File
 4. Test for Existence of PC File       16. PC File           -> S/36 File
                                        17. PC File           -> S/36 Libr Member
                                        18. PC Print File     -> S/36 Print Item


          Translation Functions                    Translation Tables
  8. S/36 File        -> PC File          20. Edit Translation Table
  9. S/36 Print Item  -> PC Print File    21. Compile Text Translation File
 10. PC File          -> S/36 File
 11. PC Print File    -> S/36 Print Item


  Enter Menu Item Number of Program to Execute
 ===>
```

Each option executes an ETU/36 procedure:

1. executes the ALLOCATE procedure.
2. executes the RENAME procedure.
3. executes the DELETE procedure.
4. executes the TESTFILE procedure.
8. executes the XLT36FIL procedure.
9. executes the XLT36PRT procedure.
10. executes the XLTPCFIL procedure.
11. executes the XLTPCPRT procedure.

13. executes the FILETOPC procedure.
14. executes the LIBRTOPC procedure.
15. executes the PRN1TOPC procedure.
16. executes the FILEFRPC procedure. -
17. executes the LIBRFRPC procedure.
18. executes the PRNTFRPC procedure.

20. executes the EDITABLE procedure.
21. executes the COMPILE procedure.

The ETU/36 procedures are described alphabetically in this chapter.

ALLOCATE Procedure

Table 3-2

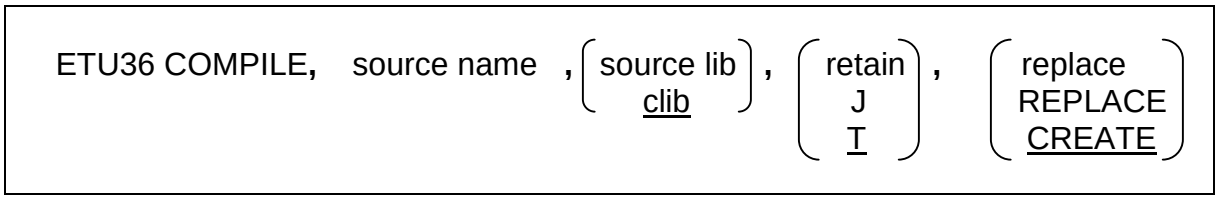
ETU36 ALLOCATE**,** workstation file name

The ALLOCATE procedure allocates a new file on the workstation.

workstation file name is the name of the workstation file to be allocated. For additional information, see "Workstation File Name" on page 4-2.

COMPILE Procedure

Table 3-3



The COMPILE procedure creates a character-oriented (*TEXT) translation file from a *TXT translation source member.

source name is the *'TEXT translation source member name. Seven members are provided in the ETU/36 library member:

E@A EBCDIC-to-ASCII (variable-length record)

E@AF EBCDIC-to-ASCII (fixed-length record)

A@E ASCII-to-EBCDIC (fixed and variable length)

A@E3 ASCII-to-EBCDIC (print translation)

Mac

E@AM EBCDIC-to-ASCII (variable-length record)

E@AFM EBCDIC-to-ASCII (fixed-length record)

A@EM ASCII-to-EBCDIC (fixed and variable length)

source lib (optional) is the name of the library containing the source member (source name) to be compiled. The default is the current library.

xlat file is the label (name) of the *TEXT translation file to hold the compiled table on the S/36 disk. This file is the table actually used in the translation process, not the source member specified above.

retain (optional) is the file retention (how long the file is to exist) for the translation file. Allowed values are T or J. Specifying T results in the file existing indefinitely (until specifically deleted), and specifying J results in the file being automatically deleted at the end of the current S/36 job.
The default value is T.

replace (optional) replaces an existing translate table previously existing on disk. Enter CREATE if this is a new table file, or REPLACE to replace an existing file. The default value is CREATE.

DELETE Procedure

Table 3-4

ETU36 DELETE**,**	workstation file name **,,,**	(notify NO YES)	

The DELETE procedure deletes a file on the workstation.

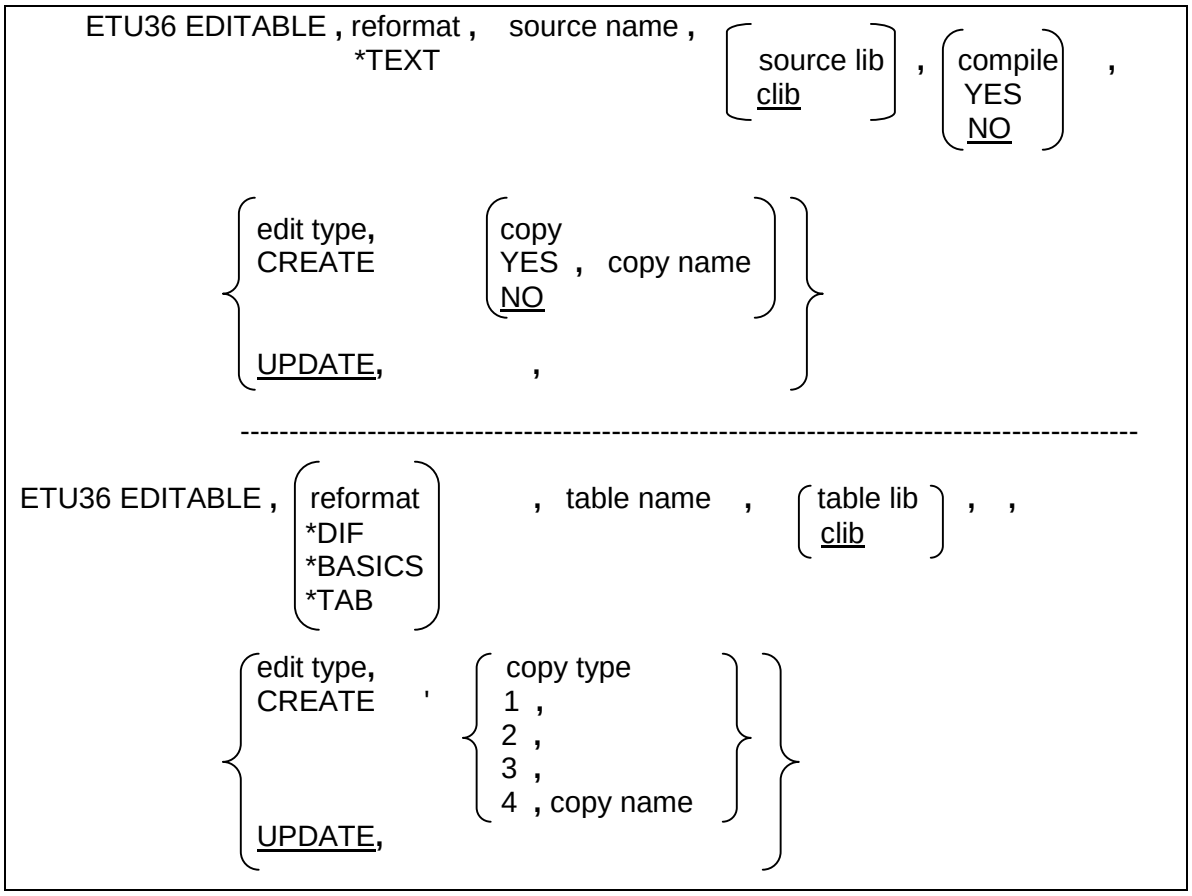
workstation file name is the name of the workstation file to be deleted. For additional information, see "Workstation File Name" on page 4-2.

notify (optional) specifies whether you should be notified with a halt message if the workstation file is not found in the directory. If NO is specified, the procedure continues without halting. When the file is not found, and if YES is specified, the procedure halts with the message "USER- 4000 File Not Found In Directory." The default value is YES.

Mac Users: The directory dialog box is unavailable for the DELETE option on the ETU36 menu.

EDITABLE Procedure

Table 3



The EDITABLE procedure edits the source members for the character-oriented translation files, and the load members for the field-oriented translation programs.

reformat specifies the type of translation file to be edited. *TEXT specifies that a character-oriented source member is to be edited. *DIF *BASICS or *TAB specifies that a field-oriented member is to be edited.

source name is the source member name of the character-oriented translation table to be edited. Seven members are provided in the ETU36 library member:

E@A EBCDIC-to-ASCII (variable-length record)

E@AF EBCDIC-to-ASCII (fixed-length record)

A@E ASCII-to-EBCDIC (fixed and variable length)

A@E3 ASCII-to-EBCDIC (print translation)

Mac

A@EM EBCDIC-to-ASCII (variable-length record)

E@AFM EBCDIC-to-ASCII (fixed-length record)

A@EM ASCII-to-EBCDIC (fixed and variable length)

It is strongly recommended that these original source members be copied into new members for modification; do not modify the originals.

table name is the name of the field-oriented translation table to be edited. This table is stored as a "load" member in the library and does not need to be compiled.

source lib (optional) is the library containing the character-oriented source member that is to be edited. The default value is the current library.

table lib (optional) is the library containing the field-oriented source member that is to be edited. The default value is the current library.

compile (optional) YES specifies that the operator is to be prompted to compile the character-oriented source member, after editing, into the S/36 disk file used in the translation process. Specify NO for compilation of the source member after editing is completed in this session. Note that if the source member is not compiled during the edit process, the COMPILE procedure must be run to create the disk file. The default value is NO.

edit type (optional) specifies whether a new table is to be created (CREATE) or if an existing table member is to be updated (UPDATE). The default value is UPDATE.

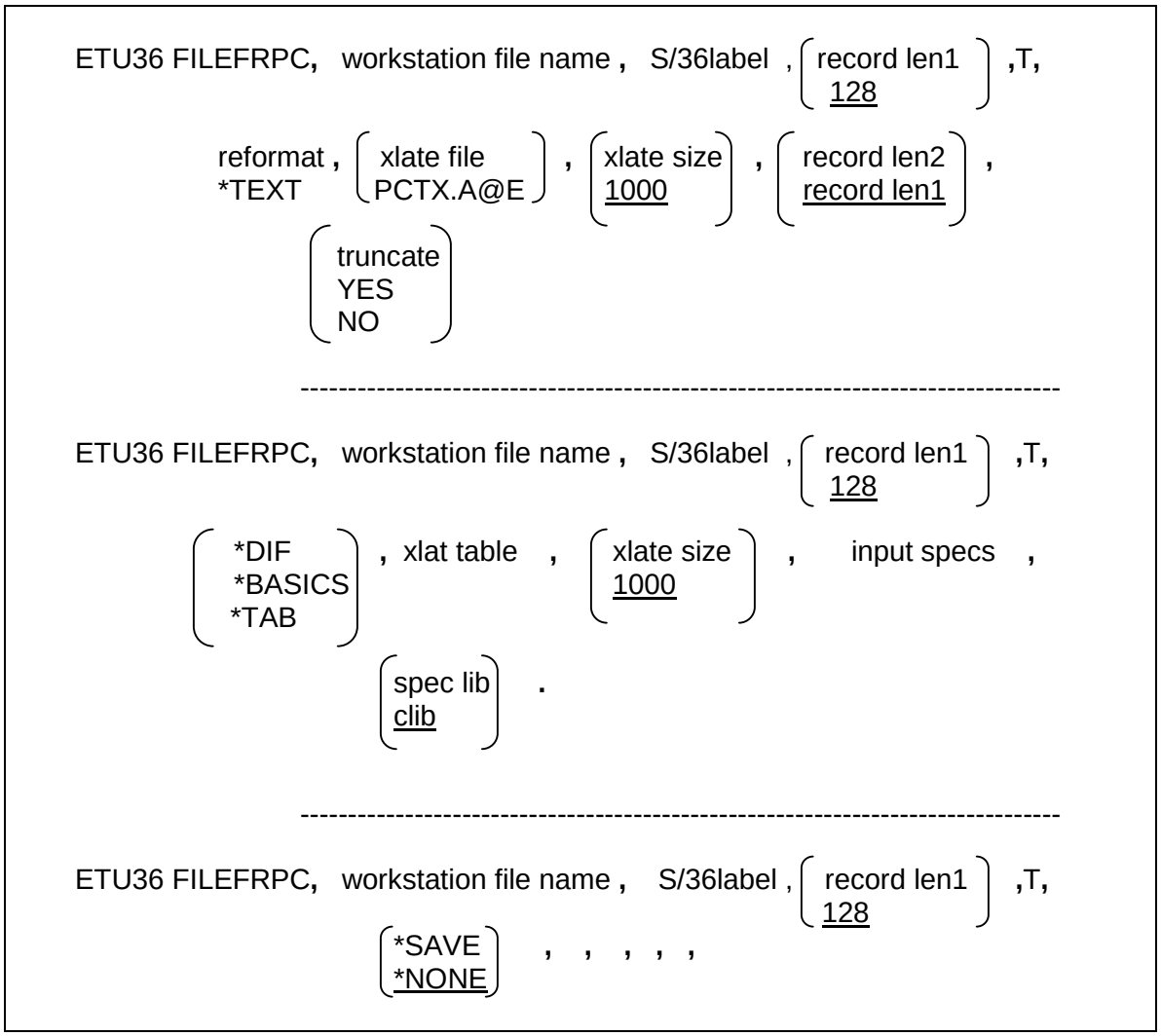
copy (optional) is YES if an existing source member is to be copied into the new source member being created. The parameter is used only if the "edit type" parameter is CREATE. Specify NO if no member is to be copied. The default value is NO.

copy type specifies what default table is to be loaded when creating a new field-oriented translation table. The value 1 loads a default EBCDIC-to-ASCII table. The value 2 loads a default ASCII-to-EBCDIC table. The value 3 loads a null table mapping all values to hex 00. The value 4 copies an existing table and uses its values as the default.

copy name is the name of the translate table to copy into the new table. This parameter is only used if the "copy type" parameter is 4. The member to be copied must exist in the library specified in the "table lib" parameter.

FILEFRPC Procedure

Table 3-6



The FILEFRPC procedure transfers a file from the workstation to the S/36, optionally translating the file into a Sf36 file format.

workstation file name is the name of the workstation file to be transferred from the workstation. For additional information, see "Workstation File Name" on page 4-2.

S/36label is the file label (name) of the S/36 file to receive the file transferred from the workstation.

record len1 (optional) is the record length of the target S/36 file ("S/36label"). This value defaults to 128.

reformat (optional) specifies the type of data translation to take place.

*__TEXT__ (or __*YES__)
indicates that the S/36 file is to be translated into an ASCII text format.

***DIF** indicates that the S/36 file is to be translated into DIF format.

***BASICS** indicates that the S/36 file is to be translated into a BASIC Sequential format. 'BASICS is invalid if you chose to use a comma as a decimal place character (instead of a period) as described in the "Installation" section of Chapter 2.

***TAB** indicates that the S/36 file is to be translated into a BASIC Sequential format with tab characters as the field delimiters.

***SAVE** permits offline storage, and moves executable programs between workstations. (See below.)

***NONE** indicates that no translation of the S/36 file is to be performed. This is the default. When using 'SAVE to move executable programs between workstations, all necessary data (including Mac data and resource forks) is transferred. Data is not translated to EBODIC, and therefore cannot be used on the host. To restore data to the workstation, use the FILETOPC procedure with reformat type 'SAVE

xlat file (optional) Is the translation table file on the S/36 disk used for character-oriented (*TEXT) translation. Translation file PCTX.A@E and PCTXA@EF are provided on the S/36 distribution diskette and are used to translate from the ASCII to the EBCDIC character set. See the EDITABLE Procedure and COMPILE procedures for more information on creating new translation table files. The default is PCTX.A@E.

MacIntosh Users: Do not use the default "xlat file" value. You must use the file PCMC.A@E.

xlat table (optional) is the translation table used in the field-oriented translation methods ('DIF, 'BASICS, 'TAB). The table ("xlat table") must exist in the same library ("spec lib") as the specifications of the fields ("input specs"). The default of BLANK uses the provided standard table coded into the ETU for ASCII-to-EBCDIC translation. See the EDITABLE Procedure and COMPILE procedures for more information on creating new translation table files.

Mac Users: Do not use the "xlat table" default value. You must use the table A@EM.

xlat size (optional) is the number of records In the temporary work file used in the translation process. The default value is 1000.

record len2 (optional) is the record length of the translation work file. The default value is the record length of the target file ("record len 1").

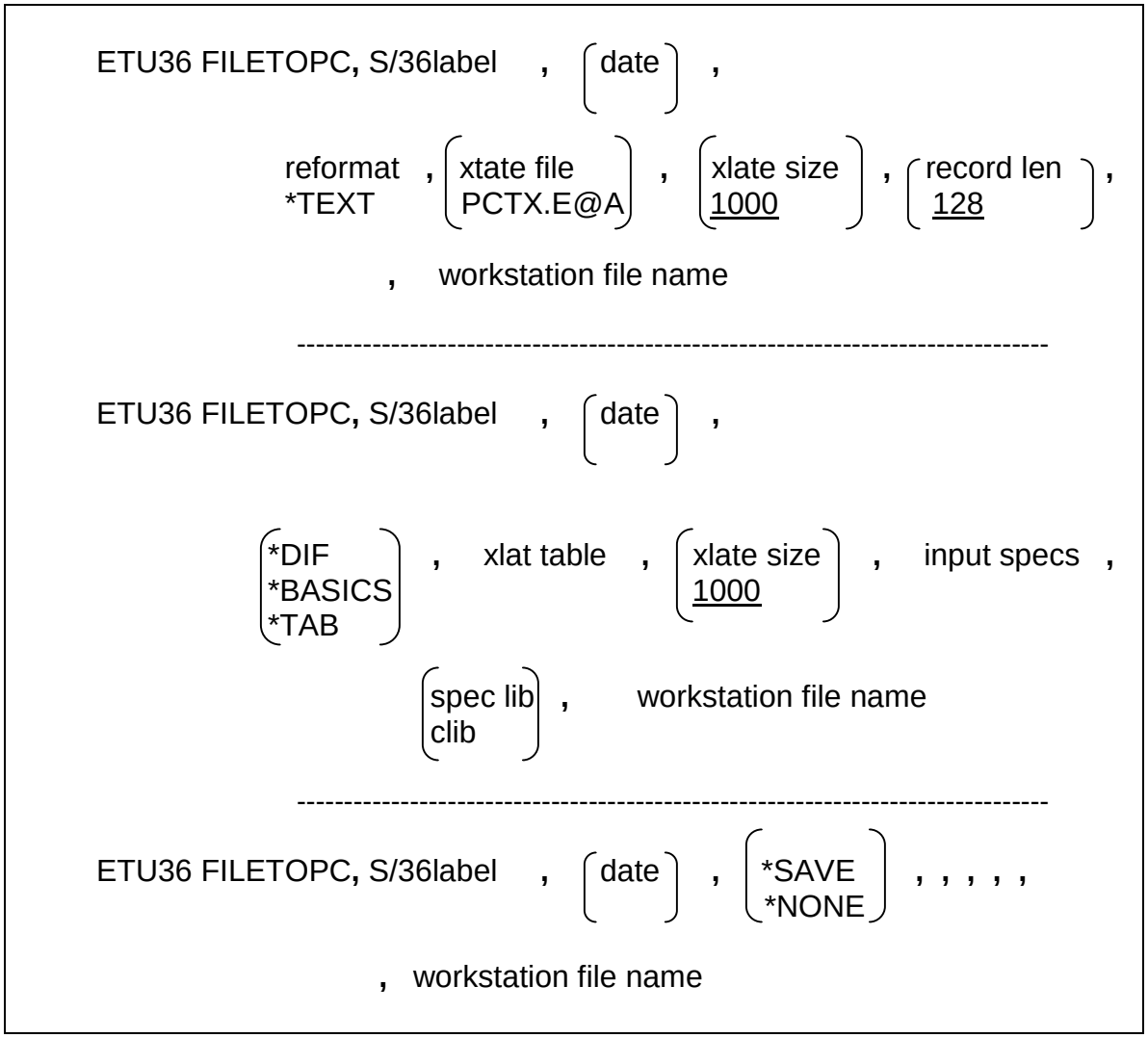
input specs is the name of the field definition member in user library "spec lib". These are standard F and I (file/input) RPG specifications, defining the file and its fields. This member is required only for the field-oriented translation methods ('DIF, 'BASICS, 'TAB). See Chapter 5 for more information on creating the F and I RPG specifications.

truncate (optional) is YES if data exceeding the record length of ‘S/36label’ is to be omitted. Specifying NO will cause a new record(s) to be written to contain the overflow. This parameter is ignored if translation is not performed. The default Is NO.

spec lib (optional) is the library containing the field definition member ("input specs"), and the translate table member ("xlat table"). If left blank this parameter defaults to the current library.

FILETOPC Procedure

Table 3-7



The FILETOPC procedure transfers a S/36 file from the Sf36 to the workstation specified, and optionally translates the file into a workstation data format.

S/36label is the label (name) of the file to be sent to the workstation.

date (optional) is the date of file "S/36label" (YMD format).

reformat (optional) specifies the type of data translation to take place.

*__TEXT__ (or __*YES__)
indicates that the S/36 file is to be translated into an ASCII text format.

***DIF** indicates that the S/36 file is to be translated into DIF format.

***BASICS** indicates that the S/36 file is to be translated into a BASIC Sequential format. 'BASICS is invalid if you chose to use a comma as a decimal place character (instead of a period) as described in the "Installation" section of Chapter 2.

***TAB** indicates that the S/36 file is to be translated into a BASIC Sequential format with tab characters as the field delimiters.

***SAVE** permits offline storage, and moves executable programs between workstations. (See below.)

***NONE** indicates that no translation of the S/36 file is to be performed. This is the default. When using 'SAVE to move executable programs between workstations, all necessary data (including Mac data and resource forks) is transferred. Data is not translated to EBODIC, and therefore cannot be used on the host. To restore data to the workstation, use the FILETOPC procedure with reformat type 'SAVE

xlat table (optional) is the translation table used in the field-oriented translation methods ('DIF, 'BASICS, 'TAB). The table ("xlat table") must exist in the same library ("spec lib") as the specifications of the fields ("input specs"). The default of BLANK uses the provided standard table coded into the ETU for ASCII-to-EBCDIC translation. See the EDITABLE Procedure and COMPILE procedures for more information on creating new translation table files.

Mac Users: Do not use the "xlat table" default value. You must use the table A@EM.

xlat size (optional) is the number of records In the temporary work file used in the translation process. The default value is 1000.

record len2 (optional) is the record length of the translation work file. The default value is the record length of the target file ("record len 1").

input specs is the name of the field definition member in user library "spec lib". These are standard F and I (file/input) RPG specifications, defining

the file and its fields. This member is required only for the field-oriented translation methods ('DIF, 'BASICS, 'TAB). See Chapter 5 for more information on creating the F and I RPG specifications.

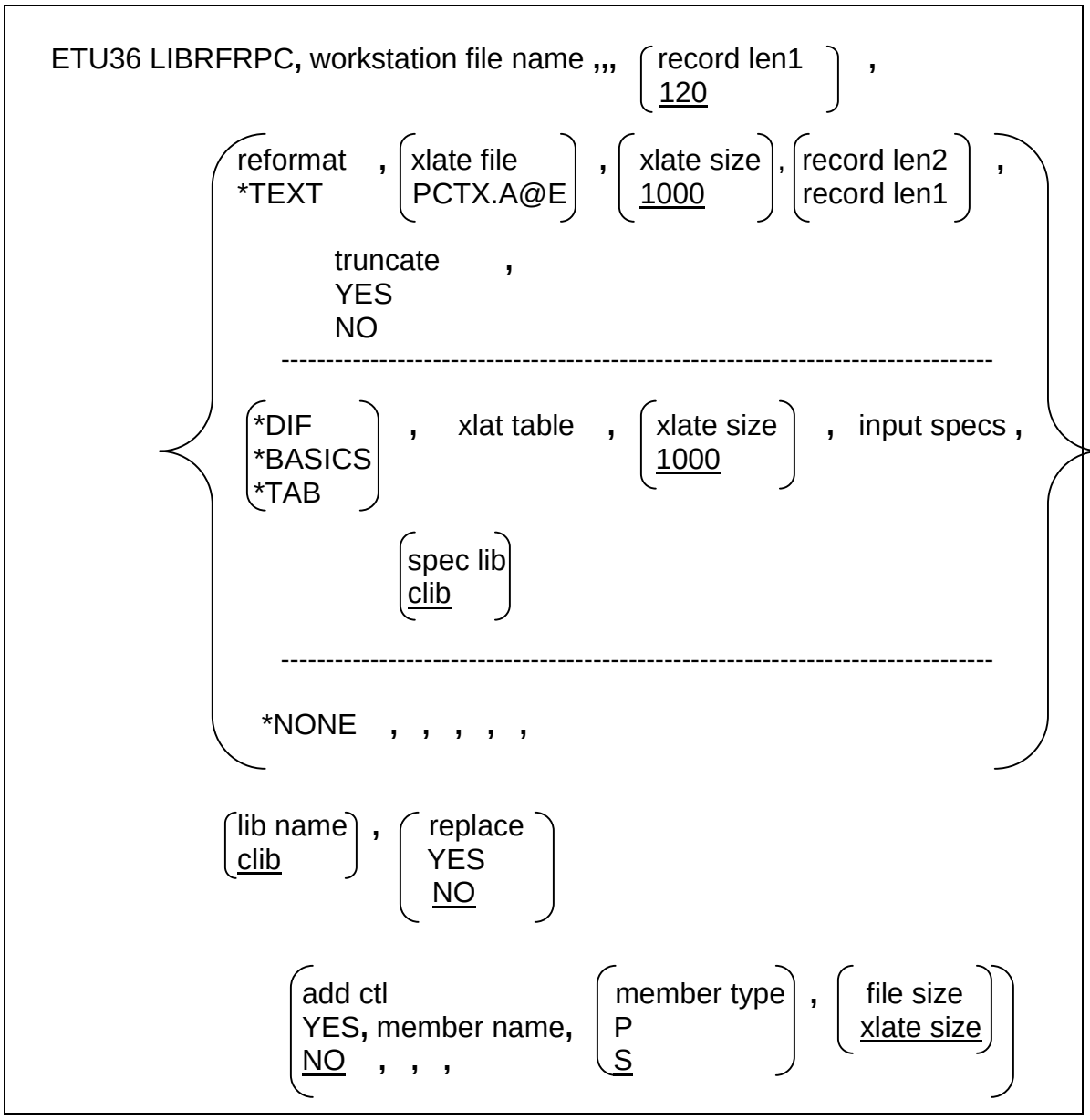
truncate (optional) is YES if data exceeding the record length of ‘S/36label’ is to be omitted. Specifying NO will cause a new record(s) to be written to contain the overflow. This parameter is ignored if translation is not performed. The default Is NO.

spec lib (optional) is the library containing the field definition member ("input specs"), and the translate table member ("xlat table"). If left blank this parameter defaults to the current library.

workstation file name is the name of the workstation file to receive the data from the S/36. For additional information, see "Workstation File Name" on page 4-2.

LIBRFRPC Procedure

Table 3-8



The LIBRFRPC procedure transfers and optionally translates a workstation file into library members on the S/36.

workstation file name is the name of the workstation file to be transferred from the workstation. For additional information, see "Workstation File Name" on page 4-2.

S/36label is the file label (name) of the Sf36 file to receive the file transferred from the workstation.

record len1 (optional) is the record length of the target Sf36 file ("S/36label"). This value defaults to 128.

reformat (optional) specifies the type of data translation to take place.

*__TEXT__ (or __*YES__)
indicates that the S/36 file is to be translated into an ASCII text format.

***DIF** indicates that the S/36 file is to be translated into DIF format.

***BASICS** indicates that the S/36 file is to be translated into a BASIC Sequential format. 'BASICS is invalid if you chose to use a comma as a decimal place character (instead of a period) as described in the "Installation" section of Chapter 2.

***TAB** indicates that the S/36 file is to be translated into a BASIC Sequential format with tab characters as the field delimiters.

***SAVE** permits offline storage, and moves executable programs between workstations. (See below.)

***NONE** indicates that no translation of the S/36 file is to be performed. This is the default. When using 'SAVE to move executable programs between workstations, all necessary data (including Mac data and resource forks) is transferred. Data is not translated to EBODIC, and therefore cannot be used on the host. To restore data to the workstation, use the FILETOPC procedure with reformat type 'SAVE

xlat file (optional) Is the translation table file on the S/36 disk used for character-oriented (*TEXT) translation. Translation file PCTX.A@E and PCTXA@EF are provided on the S/36 distribution diskette and are used to translate from the ASCII to the EBCDIC character set. See the EDITABLE Procedure and COMPILE procedures for more information on creating new translation table files. The default is PCTX.A@E.

MacIntosh Users: Do not use the default "xlat file" value. You must use the file PCMC.A@E.

xlat table (optional) is the translation table used in the field-oriented translation methods ('DIF, 'BASICS, 'TAB). The table ("xlat table") must exist in the same library ("spec lib") as the specifications of the fields ("input specs"). The default of BLANK uses the provided standard table coded into the ETU for ASCII-to-EBCDIC translation. See the EDITABLE Procedure and COMPILE procedures for more information on creating new translation table files.

Mac Users: Do not use the "xlat table" default value. You must use the table A@EM.

xlat size (optional) is the number of records In the temporary work file used in the translation process. The default value is 1000.

record len2 (optional) is the record length of the translation work file. The default value is the record length of the target file ("record len 1").

input specs is the name of the field definition member in user library "spec lib". These are standard F and I (file/input) RPG specifications, defining the file and its fields. This member is required only for the field-oriented translation methods ('DIF, 'BASICS, 'TAB). See Chapter 5 for more information on creating the F and I RPG specifications.

truncate (optional) is YES if data exceeding the record length of ‘S/36label’ is to be omitted. Specifying NO will cause a new record(s) to be written to contain the overflow. This parameter is ignored if translation is not performed. The default Is NO.

spec lib (optional) is the library containing the field definition member ("input specs"), and the translate table member ("xlat table"). If left blank this parameter defaults to the current library.

lib name (optional) is the name of the library on the S/36 to place the members from the workstation. The default is the current library.

replace (optional) specifies whether or not (YES or NO respectively) to replace an existing library member without a warning message. The default value Is NO.

add ctl (optional) specifies whether or not (YES or NO respectively) to add $MAINT utility control statements into the work file before sending the member to the library. The default value is NO.

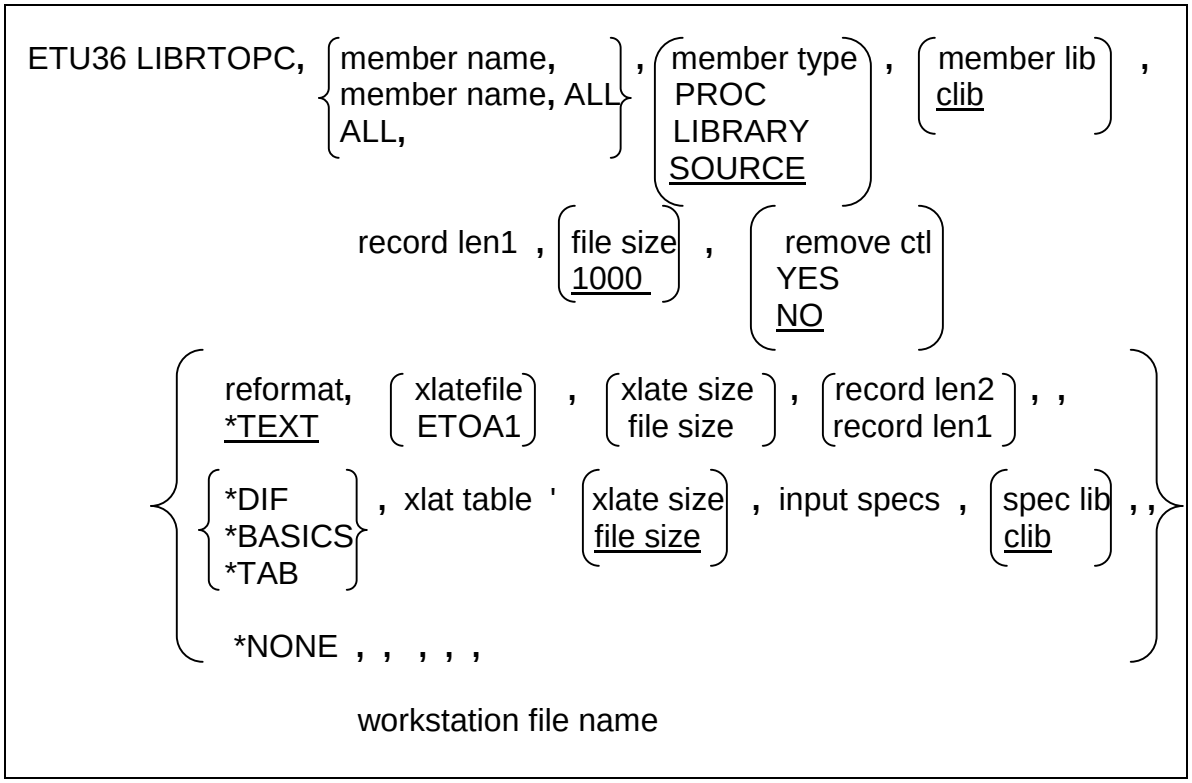
member name if "add ctl" is YES, is the library member name of the work.station data when replaced in the library (lib name).

member type (optional) if "add ctl" is YES, is the library member type to be created. Specify S for source and P for procedure. The default value is S.

file size (optional) is the work file size for the "add ctl" process. This defaults to "xlat size".

LIBRTOPC Procedure

Table 3-9



The LIBRTOPC procedure transfers a S/36 library member from the S/36 to the workstation specified, optionally translating the file into a workstation data format.

member name is the name, or partial name, of the library member(s) to be sent to the workstation.

member type (optional) is the type of the library member(s) to be extracted. Specify PROC for procedures, SOURCE for source members, or LIBRARY for both source and procedure members. The default value is SOURCE. LOAD and DUBR create only secot (8-byte) mode files.

member lib (optional) is the name of the S/36 library from which to extract the specified members. The default value Is the current library.

record len1 (optional) is the record length of the System/36 library member(s) to be sent to the workstation. This parameter defaults to 96 if SOURCE is specified for “member type", 120 if PROC or LIBRARY is specified, or 8 if LOAD or SUBR Is specified. A record length of 8 creates a sector mode file, and 40 to 120 create record mode files.

file size (optional) is the number of records in the temporary work file used in the translation process. The default value is 1000.

remove ctl (optional) specifies whether to remove the S/36 utility control (SMAINT) statements from the library member(s) before they are Sent to the workstation. Specifying YES removes the statements. The default value is NO.

reformat (optional) specifies the type of data translation to take place.

*__TEXT__ (or __*YES__) indicates that the S/36 file is to be translated into an ASCII text format.

__*DIF__ indicates that the S/36 file is to be translated into DIF format.

__*BASICS__ indicates that the S/36 file is to be translated into a BASIC Sequential format. 'BASICS is invalid if you chose to use a comma as a decimal place character (instead of a period) as described in the "Installation" section of Chapter 2.

__*TAB__ indicates that the S/36 file is to be translated into a BASIC Sequential format with tab characters as the field delimiters.

__*SAVE__ permits offline storage, and moves executable programs between workstations. (See below.)

__*NONE__ indicates that no translation of the S/36 file is to be performed. This is the default. When using 'SAVE to move executable programs between workstations, all necessary data (including Mac data and resource forks) is transferred. Data is not translated to EBODIC, and therefore cannot be used on the host. To restore data to the workstation, use the FILETOPC procedure with reformat type 'SAVE

xlat file (optional) Is the translation table file on the S/36 disk used for character-oriented (*TEXT) translation. Translation file

PCTX.A@E and PCTXA@EF are provided on the S/36 distribution diskette and are used to translate from the ASCII to the EBCDIC character set. See the EDITABLE Procedure and COMPILE procedures for more information on creating new translation table files.

Mac Users: Do not use the default "xlat file" value. You must use either PCMC.E@A for variable length records or PCMCE@AF for fixed length records.

xlat table (optional) is the translation table used in the field-oriented translation methods ('DIF, 'BASICS, 'TAB). The table ("xlat table") must exist in the same library ("spec lib") as the specifications of the fields ("input specs"). The default of BLANK uses the provided standard table coded into the ETU for ASCII-to-EBCDIC translation. See the EDITABLE Procedure and COMPILE procedures for more information on creating new translation table files.

Mac Users: Do not use the "xlat table" default value. You must use the table A@EM.

xlat size (optional) is the number of records In the temporary work file used in the translation process. The default value is 1000.

record len2 (optional) is the record length of the translation work file. The default value is the record length of the target file ("record len 1").

input specs is the name of the field definition member in user library "spec lib". These are standard F and I (file/input) RPG specifications, defining the file and its fields. This member is required only for the field-oriented translation methods ('DIF, 'BASICS, 'TAB). See Chapter 5 for more information on creating the F and I RPG specifications.

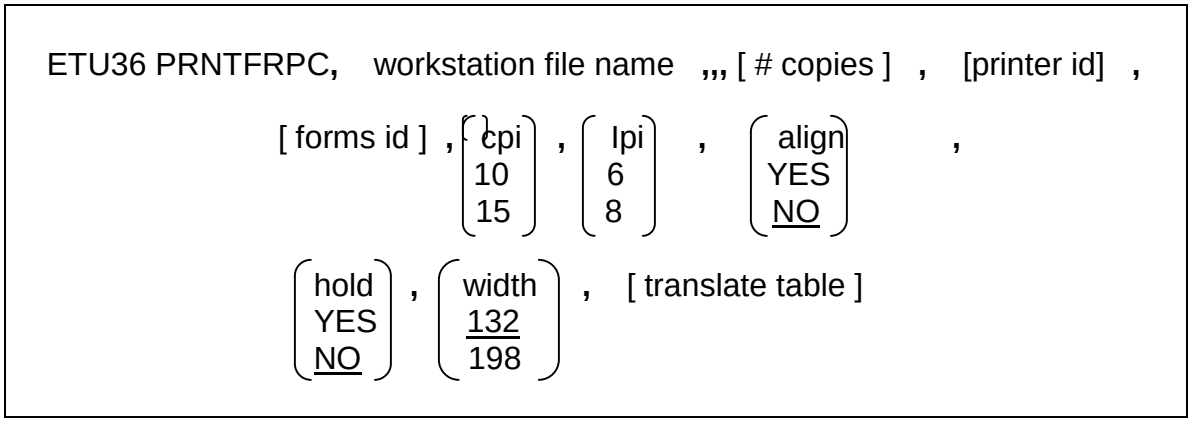
truncate (optional) is YES if data exceeding the record length of ‘S/36label’ is to be omitted. Specifying NO will cause a new record(s) to be written to contain the overflow. This parameter is ignored if translation is not performed. The default Is NO.

spec lib (optional) is the library containing the field definition member ("input specs"), and the translate table member ("xlat table"). If left blank this parameter defaults to the current library.

workstation file name is the name of the workstation file to be transferred from the workstation. For additional information, see "Workstation File Name" on page 4-2.

PRNTFRPC Procedure

Table 3-10



The PRNTFRPC procedure transfers and translates a PC-DOS file and place it on the S/36 print spool.

Mac Users: Do not run this procedure. It is intended for use only with PC-DOS.

workstation file name is the name of the workstation file to be transferred from the workstation. For additional information, see "Workstation File Name" on page 4-2.

 # copies (optional) is the number of copies of the report to print on the S/36. The default is one copy.

printer id (optional) is the *S/36* printer id to which the report is to be sent. If left blank, the entry will default to the value specified in the configuration of the workstation on which the PRNTFRPC procedure is executed.

forms id (optional) is the forms number (name) on which to print the report. If left blank, the entry will default to the value specified in the configuration of the workstation on which the PRNTFRPC procedure is executed.

cpi (optional) is the characters-per-inch format for the report. If left blank, the entry will default to the value specified in the configuration of the workstation on which the PRNTFRPC procedure is executed.

lpi (optional) is the lines-per-inch format for the report. If left blank, the entry will default to the value specified in the configuration of the workstation on which the PRNTFRPC procedure is executed.

align (optional) specifies whether a forms alignment halt will be issued on the printer's console for this report. The default is NO.

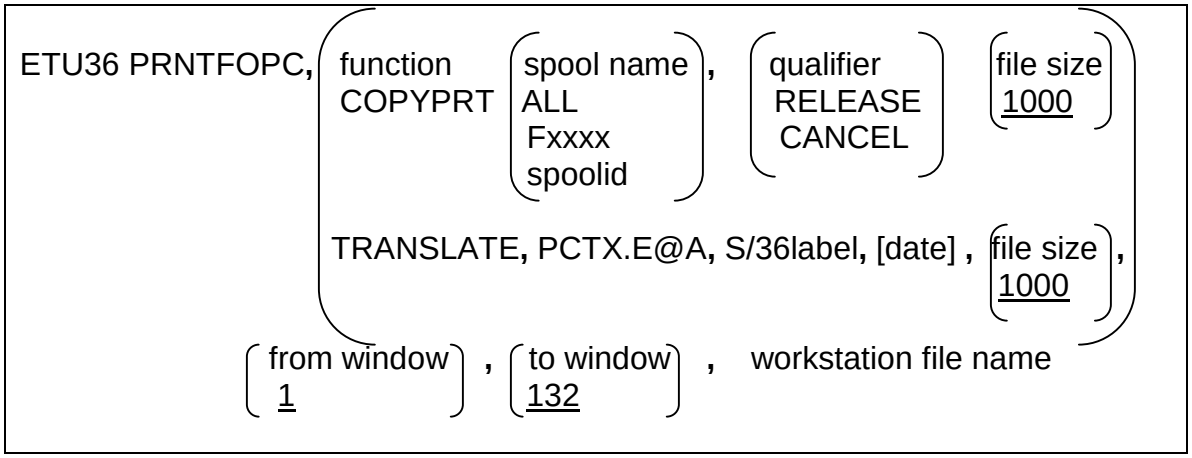
hold (optional) specifies whether the report will be "held" on the print spool until the operator releases it for printing. The default is NO.

width (optional) specifies the width in columns of the file to be printed. Width may be 132 (the default) or 198.

translate table (optional) specifies the name of the printer translation table you wish to use. The default is PCTXA@E.

PRNTTOPC Procedure

Table 3-11



The PRNITOPC procedure translates and transfers print spool reports from the S/36 to the workstation.

Mac Users: Do not run this procedure. It is intended for use only with PC-DOS.

function has possible values of COPYPRT or TRANSLATE. Selecting COPYPRT indicates that the SSP COPYPRT procedure will be executed and that the print report(s) will be extracted from the S/36 print spool prior to translation into a workstation print format. Selecting TRANSLATE specifies that an existing COPYPRT file will be translated into the workstation print format.

spool name is the spool name (id) of the print report(s) on the S/36 print spool. Valid entries are the spool id (in the form SP*xxxx*), the forms type to be extracted (in the form F*xxxx*), or ALL to extract all inactive print spool items. Examine the COPYPRT procedure in the manual *System Reference for the S/36 Environment* for more Information on these entries.

qualifier (optional) is the action to be taken regarding the copied S/36 spool items after the print reports have been copied. RELEASE specifies that the copied print reports are released for printing and will print at the next opportunity. CANCEL specifies that the copied print spool items are no longer needed and are to be deleted from the print spool. If omitted, no action will be taken on the spool item.

PCTXE@A (optional) the translation table file on the S/36 disk used for translating print spool files from the S/36 to a format recognized by the workstation.

S/36label is the label (name) of the S/36 COPYPRT print file existing on the S/36 disk that is to be translated into the workstation print format.

date (optional) is the date of the S/36 COPYPRT file "S/36label" (YMD format).

file size (optional) is the number of records in the temporary work file used in the translation process. The default value is 1000.

from window (optional) is the print column position of the report(s) on which to start translation. Any number from 1 to 198 is valid. The default value is 1.

to window (optional) is the print column position of the report(s) on whIch to end translation. Any number from 1 to 198 (must be equal to or larger than the number specified in "from window") is valid. The default value is 132.

workstation file name is the name of the workstation file to be transferred from the workstation. For additional information, see "Workstation File Name" on page 4-2.

RENAME Procedure

Table 3-12

ETU36 RENAME**,** workstation file name **,,,** new file name

The RENAME procedure changes the filename of a workstation file.

workstation file name is the name of the workstation file to be transferred from the workstation. For additional information, see "Workstation File Name" on page 4-2.

You may Include the DOS pathname In the file specification (e.g., C:\TFST\MYFILETXT) If not specified, it defaults to the current default directory on your workstation.

new file name is the new name to be given to the workstation file.

NOTE: The new file name plus extension should be of the form XXXXXXXX. YYY where XXXXXXXX Is a file name of up to eight characters, and . YYY is an optional extension of up to three characters.

Mac Users: The directory dialog box is unavailable for the RENAME option on the ETU36 menu.

TESTFILE Procedure

Table 3-13

ETU36 TESTFILE**,** workstation file name **,,,** (lda offset / NOTIFY)

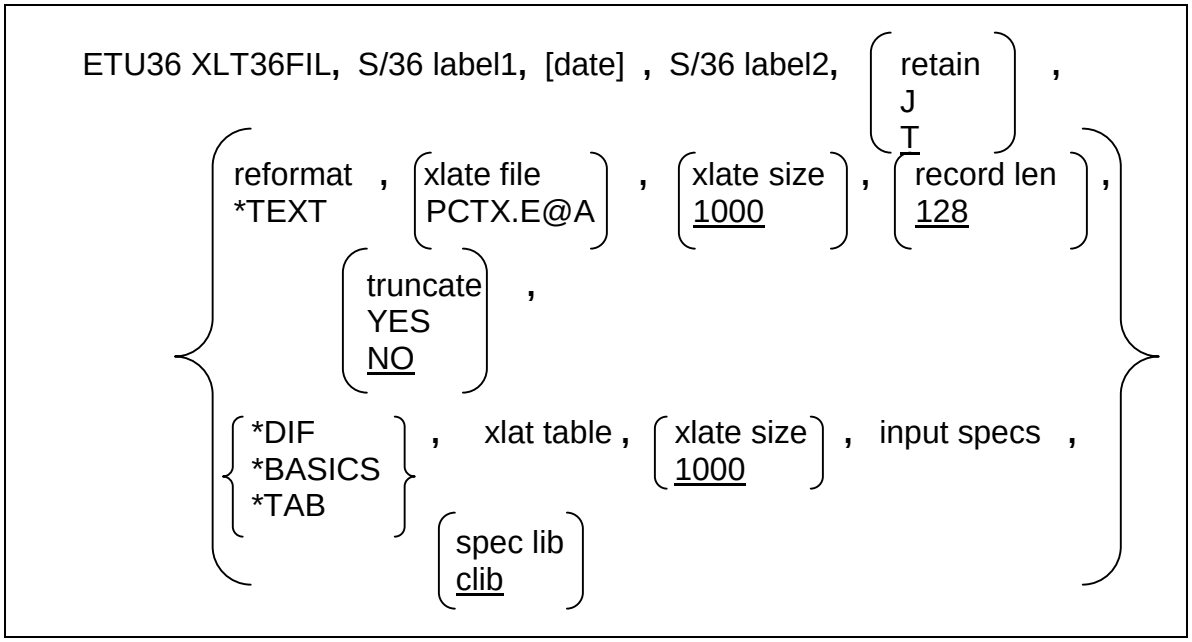
The TESTFILE procedure tests for the presence of the specified file and returns the number of bytes (characters) in the file. Depending on the "lda offset" parameter, the results of the test will be either given to the operator as a message, or placed in the Local Data Area (LDA) for later use by the programmer in determining the next step to process in the application program.

workstation file name is the name of the workstation file to be transferred from the workstation. For additional information, see "Workstation File Name" on page 4-2.

lda offset (optional) has two possible values: NOTIFY and LDA position. If NOTIFY is specified, the operator will be sent a message with the results of the test. If a number is specified, it indicates the starting position in the LDA (Local Data Area) where the results of the test will be placed. The results of the test are 12 bytes long and will start at the specified LDA offset, anywhere from ito 501. The first 4 bytes are the return code. If the file is found, the return code is 0000. Any other return code indicates either an error or that the file was not found. The remaining 8 bytes contain the workstation file size, if the file is found. The default is NOTIFY.

XLT36FIL Procedure

Table 3-14



The XLT36FIL procedure performs translation of a file residing on the S/36 disk into a workstation data format. This procedure may be run from the job queue (with JOBQ), or released from the workstation (with EVOKE).

S/36label1 is the label (name) of the S/36 file to be translated.

date (optional) is the date of the S/36 file to be translated (YMD format).

S/36labe12 is the label (name) of the file that will contain the results of the translation process (the workstation-formatted data). This name cannot already be on the S/36 disk when this procedure is called.

retain (optional) is the tile retention (how long the file is to exist) for the translated "S/361abe12". Allowed values are T or J. Specifying T results in the existing indefinitely (until specifically deleted), and specifying J will in the file being automatically deleted at the end of the current S/36 job. The default value is T.

reformat (optional) specifies the type of data translation to take place.
*__TEXT__ (or __*YES__)
indicates that the S/36 file is to be translated into an ASCII text format.

***DIF** indicates that the S/36 file is to be translated into DIF format.

***BASICS** indicates that the S/36 file is to be translated into a BASIC Sequential format. 'BASICS is invalid if you chose to use a comma as a decimal place character (instead of a period) as described in the "Installation" section of Chapter 2.

***TAB** indicates that the S/36 file is to be translated into a BASIC Sequential format with tab characters as the field delimiters.

***SAVE** permits offline storage, and moves executable programs between workstations. (See below.)

***NONE** indicates that no translation of the S/36 file is to be performed. This is the default. When using 'SAVE to move executable programs between workstations, all necessary data (including Mac data and resource forks) is transferred. Data is not translated to EBODIC, and therefore cannot be used on the host. To restore data to the workstation, use the FILETOPC procedure with reformat type 'SAVE

xlat file (optional) is the translation table file on the S/36 disk used for character-oriented translation (*TEXT). Translation file PCTX.E@A is provided on the S/36 distribution diskette and can be used to translate from the EBCDIC to the ASCII character set in variable length records. Translation file PCTXE@AF is provided on the S/36 distribution diskette and can be used to translate from the EBCDIC to the ASCII character set in fixed length records. See the EDITABLE Procedure and COMPILE procedures for more information on creating new translation table files. The default
value is PCTX.E@A.

Mac Users: Do not use the "xlat table" default value. You must use the table A@EM.

xlat table (optional) is the translation table used in the field-oriented translation methods ('DIF, 'BASICS, 'TAB). The table ("xlat table") must exist in the same library ("spec lib") as the specifications of the fields ("input specs"). The default of BLANK uses the provided standard table coded into the ETU for ASCII-to-EBCDIC translation. See the EDITABLE Procedure and COMPILE procedures for more information on creating new translation table files.

xlat size (optional) is the number of records In the temporary work file used in the translation process. The default value is 1000.

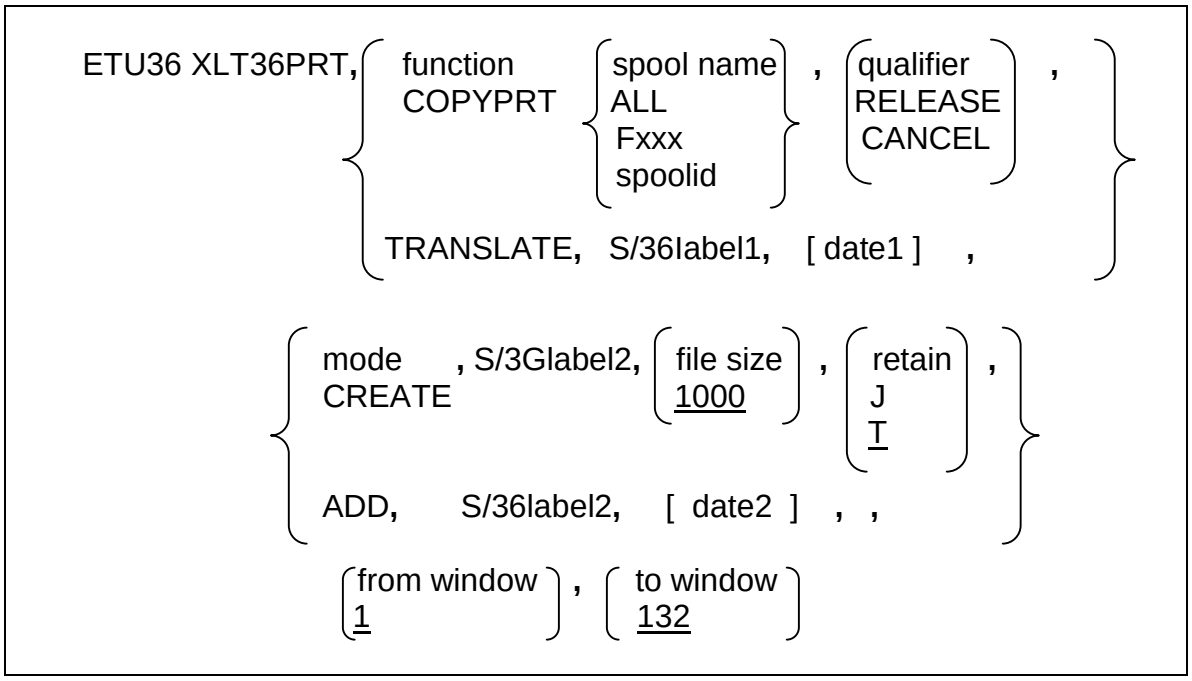
record len (optional) is the record length of the output file (“S/36label2”). The default value is 128.

input specs is the name of the field definition member in user library "spec lib". These are standard F and I (file/input) RPG specifications, defining the file and its fields. This member is required only for the field-oriented translation methods ('DIF, 'BASICS, 'TAB). See Chapter 4, "File Transfer Facility," for more information on creating the F and I RPG specs

spec lib (optional) is the library containing the field definition member ("input specs"), and the translate table member ("xlat table"). If left blank this parameter defaults to the current library.

XLT36PRT Procedure

Table 3-15



The XLT36PRT procedure translates Sf36 print spool items into a S/36 file in the workstation print format

function has possible values of COPYPRT or TRANSLATE. Selecting COPYPRT indicates that the SSP COPYPRT procedure will be executed and that the print reports will be extracted from the S/36 print spool prior to translation into a workstation print format. Selecting TRANSLATE specifies that an existing COPYPRT file will be translated into the workstation print format.

spool name is the spool name (id) of the print reports on the S/36 print spool. Valid entries are the spool Id (In the form SP*xxxx*), the forms type to be extracted (in the form F*xxxx*), or ALL to extract all print reports. Examine the COPYPRT procedure in the Sf36 reference manual for more information on these entries.

S/36label1 is the label (name) of the S/36COPYPRT print file existing on the S/36 disk that is to be translated into the PC print format.

qualifier (optional) is the action to be taken regarding the copied Sf36 spool items after the print reports have been copied. RELEASE specifies that the copied print reports are released for printing and will print at the next opportunity. CANCEL specifies that the copied spool items are no longer needed and are to be deleted from the

print spool. Blank is the default, meaning "Do not change the print status."

date-1 (optional) is the date of the Sf36 COPYPRT file "SI36labell".

mode (optional) specifies whether the target translation file is to be created by this procedure, or if an existing file is to be changed. Specify CREATE to create a new file with the name specified In the parameter "S/36label2". ADD to add the print reports to the existing file "S/361abe12". The default value is CREATE.

S/361abe12 is the label (name) of the file to receive the translated data.

file size (optional) is the number of records in the temporary work file used in the translation process. The default value is 1000.

date-2 (optional) is the date of the *S/36* file to be added to "S/36label2".

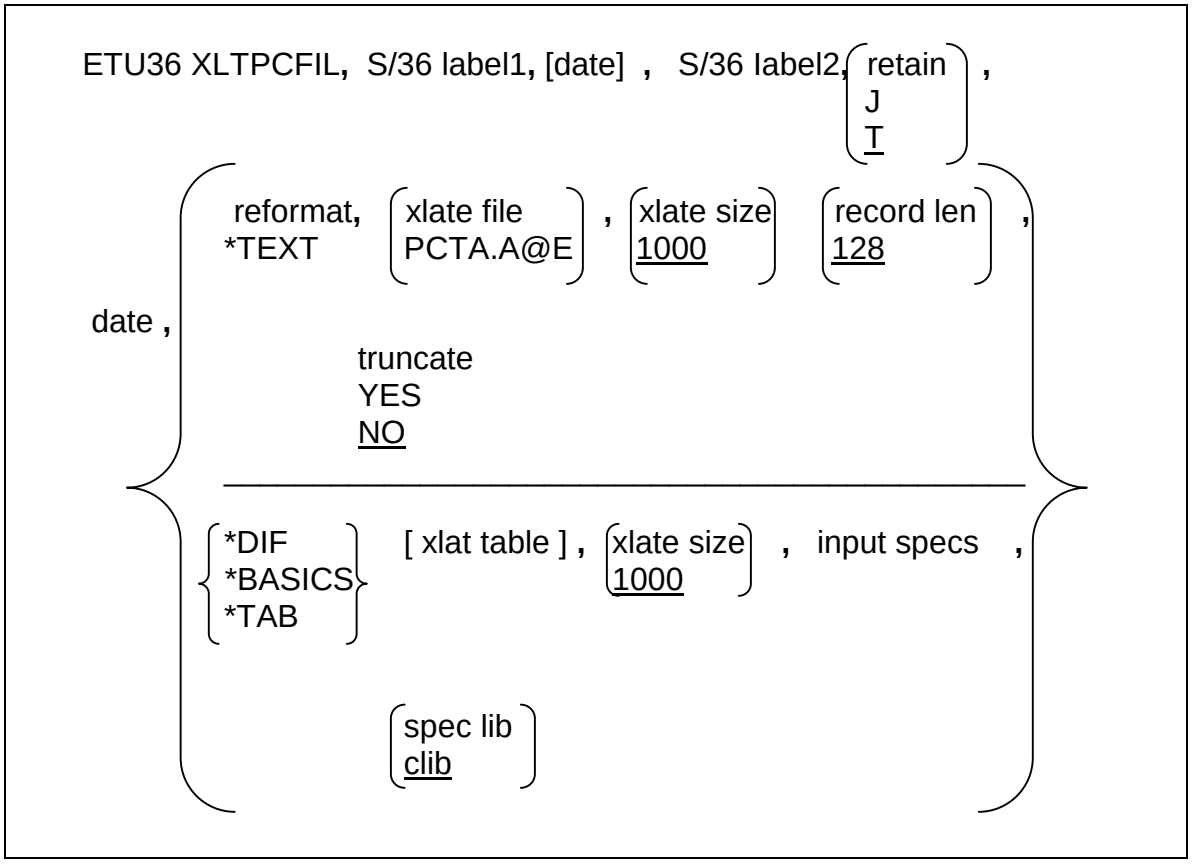
retain (optional) is the file retention (how long the file Is to exist) for the translated "S/361abe12". Allowed values are Tor J. Specifying T will result in the file existing Indefinitely (until specifically deleted), and specifying I will result in the file being automatically deleted at the end of the current S/36 job. The default value is T.

from window (optional) is the print column position of the report(s) on which to start translation. Any number from 1 to 198 is valid. The default is 1.

to window (optional) is the print column position of the report(s) on which to end translation. Any number from 1 to 198 Is valid (it must be equal to or larger than the number specified In "from window"). The default value is 132.

XLTPCFIL Procedure

Table 3-16



The XLTPCFIL procedure translates a previously transferred workstation data file into a S/36 formatted data file.

S/36label1 is the name of workstation data file residing on the S/36 that is to be reformatted for the S/36.

date (optional) is the date of the "S/36labell" file (YMD format).

S/361abe12 is the label (name) of the file that will contain the results of the translation process. This name cannot already be on the S/36 disk when this procedure is called.

retain (optional) is the file retention (how long the file is to exist) for the translated file "S/361abe12". Allowed values are T or J. Specifying T results in the file existing indefinitely (until specifically deleted), and specifying J will result in the file being automatically deleted at the end of the current S/36 job. The default value is T.

reformat (optional) specifies the type of data translation to take place.

*__TEXT__ (or __*YES__)
indicates that the S/36 file is to be translated into an ASCII text format.

***DIF** indicates that the S/36 file is to be translated into DIF format.

***BASICS** indicates that the S/36 file is to be translated into a BASIC Sequential format. 'BASICS is invalid if you chose to use a comma as a decimal place character (instead of a period) as described in the "Installation" section of Chapter 2.

***TAB** indicates that the S/36 file is to be translated into a BASIC Sequential format with tab characters as the field delimiters.

***SAVE** permits offline storage, and moves executable programs between workstations. (See below.)

***NONE** indicates that no translation of the S/36 file is to be performed. This is the default. When using 'SAVE to move executable programs between workstations, all necessary data (including Mac data and resource forks) is transferred. Data is not translated to EBODIC, and therefore cannot be used on the host. To restore data to the workstation, use the FILETOPC procedure with reformat type 'SAVE

xlat file (optional) is the translate table used in the field-oriented translation methods (*DIF, *BASICS, *TAB). The table ("xlat table") must exist in the same library ("spec lib") as the specification of the fields ("input"). The default of blank uses the provided standard table coded into the ETU for ASCII-to-EBCDIC translation. See the EDITABLE Procedure and COMPILE procedures for more information on creating new translation table files.

Mac Users: Do not use the default "xlat table" value. You must use the table A@EM.

xlat size (optional) is the number of records in the temporary work file used in the translation process. The default value is 1000.

record len (optional) is the record length of the output file ("S/361abe12"). The default value is 128.

input specs is the name of the field definition member in operator library “spec lib". These are standard F and I (file/input) RPG specifications, defining the file and its fields. This member is required only for the field-oriented translation methods (*DIF, *BASICS, *TAB). See Chapter 4 for more information on creating F and I RPG specifications.

truncate (optional) is YES if data exceeding the record length of "S/36label2" is to be omitted. Specifying NO will cause a new record to be written to contain the overflow. This default value is NO.

spec lib (optional) is the library containing the field definition member ("input specs"), and the translate table member ("xlat table"). If left blank this parameter defaults to the current library.

XL TPCPRT Procedure

Table 3-17

ETU36 XLTPCPRT , S/36label, [date] , [# copies] , [printer id] ,

[forms id] , (cpi: 10 / 12) , (ipi: 6 / 8) , (align: YES / NO) , (hold: YES / NO)

The XLTPCPRT procedure translates a PC-DOS file and places It on the S/36 print spool.

Mac Users: Do **not** run this procedure, as unpredictable results will occur.

S/36 label is the label (name) of the previously transferred workstation print file on the Sf36 disk to be translated into Sf36 print report and placed on the Sf36 print spool.

date (optional) is the date of the workstation print file "Sf3 6label" (YMD format).

 # copies (optional) is the number of copies of the report to print on the Sf36. The default is one copy.

printer id (optional) is the S/36 printer id to which the report will be sent. If left
blank, the entry will default to the value specified in the configuration of the workstation on which the XLTPCPRT procedure is executed.

forms id (optional) is the forms number (name) on which to print the report. If left blank, the entry will default to the value specified in the configuration of the workstation on which the XLTPCPRT procedure is executed.

cpi (optional) is the characters-per inch format for the report. If left blank, the entry will default to the value specified in the configuration of the workstation on which the XLTPCPRT procedure is executed.

lpi (optional) is the lines per-inch-format for the report. If left blank, the entry will default to the value specified in the configuration of the workstation on which the XLTPCPRT procedure is executed.

align (optional) specifies whether a forms alignment halt will be issued on the printer's console for this report. The default is NO.

hold (optional) specifies whether the report will be "held" on the print spool until the operator releases it for printing. The default is NO.

Chapter 4

FILE TRANSLATION FACILITY

Translation Concepts

The data formats of files on the S/36 are considerably different from those on PC workstations. On the S/36:

* Files are stored in fixed-length records

* Field definitions are always maintained external to the file

* Data is stored In an EBCDIC character format.

On the workstation:

* Files are stored in fixed or variable record length

* Multiple field definition formats exist:

  Internal (fields defined as part of the data)
  External (fields defined outside of the file)
  No field definitions (character data only)

* Data is stored in an ASCII character set

To accommodate these differences, the ETU must translate file data between ASCII and EBCDIC, and reformat the data between the S/36 file format and the various workstation formats.

Two basic approaches are taken when translating data between the two systems: character-oriented translation and field-oriented translation.

Character-Oriented Translation

In character-oriented translation, each character (or string of characters) stands alone and is translated on an individual character (or string) basis. The placement or position of that character or string In a record is of no significance.

The *TEXT translation is character-oriented. Packed and binary formats cannot be supported with this type of translation.

Field-Oriented Translation

Data translated using the field-oriented method must be defined or separated into fields corresponding to the position the data occupies In the record. A source member must define the position and format of each field used in the file being translated. After the field has been interpreted, the data is translated on a single-character basis.

The *DIF, *BASICS, and *TAB translation formats are field-oriented.

Translation Formats

ASCII Text (*TEXT) Format

The ASCII TEXT format, sometimes called the print format on the workstation, has the following specifications:

* Variable record length

* End-of-record and end-of-file markers

* No field designation.

ASCII TEXT translation is character-oriented. Single characters or character strings (up to 9 bytes in length) are matched, as found In the input file, to entries in the Translation Table File. If an entry and corresponding translation is found In the Translation Table File, that translation is placed directly into the specified output file. If the entry Is found in the table, but no translation is associated with it, no entry is made in the output file (effectively stripping the input string). If no entry is found in the Translation Table File, the input character is placed without change into the output file.

NOTE: Because the Input characters are matched directly to the entries in the translation table file, all input must be in an alphanumeric format (no packed or binary Information) when translating into an ASCII TEXT format.

BASIC Sequential (*BASICS) Format

BASIC Sequential format is one standard method of storing data on the workstation. The following specifications apply to BASIC Sequential files:

- Variable record length

- End-of-record and end-of-file markers

- Fields are designated in the following way:

All fields are separated by commas.
Alphabetic data has quote marks around the field to distinguish from numeric.

The translation of a file to or from the Basic Sequential format is field-oriented and therefore will support any type of data storage on the System/36:

Alphanumeric
Zoned Decimal
Packed
Binary

In order to support the above-named formats, the translation utility must have the same field definitions as the file being transferred. The fields may be defined through the normal F and I specifications associated with that file name. See "Field Oriented Translations" later in this chapter, and the IBM manual *System/36 System Reference Manual SC2Z 9020* for information on creating F and I specifications.

TAB (*TAB) Format

*TAB format Is the same as *BASICS except that tab characters are used as delimiters instead of commas.

DIF (*DIF) Format

*DIF format is a standard method of storing data on the PC which was developed for use on a spreadsheet product. The DIF format Is a unique format based on vectors and tuples (fields and records). A partial DIF file listing follows:

```
TABLE
0,1
“."
VECTORS
0,0006
“.”
TUPLES
0,000000012
DATA
0,0
“.”
-1,0
BOT
0,528365
V
1,0
“MR. JOHN SMITH”
```

The translation of a file to or from the DIF format is field-oriented and therefore will support any type of data storage on the System/36:

Alphanumeric
Zoned Decimal
Packed
Binary

In order to support the above-named formats, the translation utility must have the same field definitions as the file being transferred. The fields may be defined through the normal F and I specifications associated with the specified file name. See "Field Oriented Translations" later in this chapter, or the IBM manual *System/36 System Reference Manual, SC21-9020* for information on creating F and I specifications.

Character-Oriented Translation

The source for character-oriented translation tables is kept as a library source member. Seven source and seven compiled versions are provided in the library file on the ETU distribution diskette. Table 4-1 lists the source members along with their compiled versions and translation functions.

Table 4-1 * Translate Table Source Record Format

Positions	Functions
1-18	FROM HEX (or special function)
19-36	TO HEX
37-37	Translate before/after-performing control (B/A)
38-39	Pad character (B=Blank, N=Null, or 2-byte hex character)
40-41	End-of-record (EOR) action for ASCII-to-EBCDIC translation
42-43	End-of-file (EOF) action for ASCII-~o-EBCDIC translation
44-80	Comment
81-96	Not used

The user may have special requirements that are not provided for by the supplied default tables. In that event, the tables can be modified for that special situation. Before the translation programs can use the modified source members, they must be compiled by using the COMPILE procedure provided by the ETU.

The character-oriented translation program is controlled by the compiled, user-definable translation tables that direct the generation of output data based upon sequences of characters (strings) encountered in the input data. Defined strings of I to 9 characters in length are used as scan arguments during translation, with longer strings taking precedence. When a defined

string is encountered in the input data stream, any output characters associated with the string are replaced in the output data stream.

If an input character exists in the translation table, but has no corresponding translation associated with it, the input character is effectively "stripped" (or bypassed) from the output. If data encountered In the input data stream has not been defined in the translate table, the data is moved "as is" into the output data stream. Although this scheme allows for the removal/insertion of control code sequences as well as one-for-one character code conversions, the data being operated upon must be in character format.

NOTE: Binary, floating-point, and packed decimal data will be improperly translated unless first converted into character format by user-written programs.

Additional translation functions are provided for altering record formats (fixed vs. variable length).

In support of EBCDIC-to-ASCII translation, the ETU provides the following operations:

- Scanning a record from right to left to remove trailing blanks or nulls to create variable length ASCII TEXT files.
- Inserting end-of-record (BOR) and end-of-file (EOF) characters for variable length ASCII TEXT files (typically CR/LF and hex IA).

In support of ASCII-to-EBCDIC translation, the ETU performs the following operations:

- Conversion of the transferred data into fixed length records.
- Padding records with a specified character upon encountering a specific character sequence.
- Record padding Is required in a conversion from variable length to fixed length records.
- Detecting an end-of-file (EOF) character (typically hex lA) and ending the translation process.

Translate Table Source Records

The character-oriented translate table has several divisions. They are:

- Scan characters for searching input data.
- Replace characters to place in the output stream.
- Action to be taken before or after a scan/replace. This is called control action, and includes forcing end-of-record (EOR) or end-of-file (EOF) and padding. This control action is performed only when the scan characters are found in the data to be translated.
- Statements that specify the characters to be inserted upon EOR and EOF for a translation done from a fixed-length S/36 record to a variable-length workstation record.
- Statements to specify whether the ASCII-to-EBCDIC translation is to scan from right to left for the first non-blank or non-null character. This statement also doubles as the default pad character for the EBCDIC-to-ASCII translation process.

The format of the translate table source record is shown in Table 4-2, with explanations of the functions following.

Table 4-2

Workstation Type	Source Member	Compiled Version	Translation Function
PC-DOS	A@E	PCTX.A@E	ASCII-to-EBCDIC translation
	E@A	PCTX.E@A	EBCDIC-to-ASCII variable record length translation
	A@E3	PCPR.A@E3	ASCII-to-EBCDIC print translation
	E@AF	PCTXE@AF	EBCDIC-to-ASCII fixed record length translation
Mac	A@EM	PCMC.A@EM	ASCII-to-EBCDIC translation
	E@AM	PCMC.E@AM	EBCDIC-to-ASCII variable record length translation

FROM HEX Is from 1 to 9 characters expressed in hexadecimal representation for which the input data will be scanned. When found, this data is replaced by the TO HEX characters, if any. Consult Appendix A for hexadecimal values for both ASCII and EBCDIC character sets. Hexadecimal data uses the characters 0 through 9 and A through F. Examples of hexadecimal data are 01, AF, 7E, F9, FF. The characters A through F must be in upper case.

The FROM HEX table size limits are shown in Table 4-3:

Table 4-3 * FROM HEX Table Size Limits

Number of FROM Characters	Maximum Number of Table Entries
1	256
2	50
3	50
4	40
5	30
6	20
7	15
8	15
9	15

For example, the table above Indicates that a translate table can contain a maximum of 20 "From" entries with a string length of 6 bytes.

TO HEX is from 0 to 9 characters (depending again on the type of file translation) expressed in hexadecimal. Upon finding the corresponding FROM HEX, the TO HEX characters are written to the output data.

Translate Before/After (labeled "XBA" on the compiler report) determines when the translation is performed relative to performing control action. A "B" specifies translate before control, and "A" specifies translate after control. The default is "B". Typically this is used to force padding and end-of-record processing when a CR/LF sequence is found In an ASCII-to-EBCDIC translation, An example of the source statement for this is:

FROM HEX	XBA	EOR	Comment
0D0A	B	Y	If CR/LF then pad and write

Pad Character contains the value "B", for blank (hex 40), or "N" for null (hex 00), or a hexadecimal representation of a selected pad character. If a pad character is not specified, and the utility needs to pad a record, the character specified In the 'TRAIL statement will be used. If that statement is not used, padding is done with blanks.

End-of-Record Action specifies the end of the current record being translated upon detection of the FROM HEX characters (used in ASCII-to-EBCDIC translation). it can be either a "Y" or a value encoded “*1” through "'*9". A “Y” specifies that the record currently being translated will be written, with padding performed as specified. The specifications "'1" through "'9" allow the concatenation of additional "TO HEX" characters. For example:

FROM HEX	EOR	Comment
0D0A	Y	If CR/LF, write record

Another example, writing AAAB before padding with nulls:

FROM HEX	TO HEX	EOR	Comment
0D0A		*1	If CR/LF
*1	C1 C1 C1 C2	Y	Then write AAAB

End-of-File Action identical to end-of-record action with exception that this function completes translation and terminates the program.

Comments are printed in the translation table’s compiler listing.

There are five special records for the translate table compiler. These records define what action is to be taken for end-of-record and end-of-file for EBCDIC-to-ASCII translation. On ASCII-to-EBCDIC translations, the special records determine if trailing blanks or nulls should be omitted. The *1 to *9 records specify additional "TO HEX" characters end-of-record and file processing. The five special records are defined as follows:

***EOR statement** - If *EOR is coded in the FROM HEX area (positions 1-4), this statement will define the action to be taken for a EBCDIC-to- ASCII translate upon reaching the end of the text In the record currently being translated. Normally, this statement would be used to insert a CR/LF (hex 0D0A). If the *TRAIL statement has been specified, the end of record condition Is met when the remainder of the input record being processed contains only the character specified in the TO HEX field of the *TRAIL statement.

***EOF statement** - This statement defines the action to be taken end of file on the file being translated. This can be used to write the end-of-file character hex lA in the record before closing the file to be sent to the workstation with the EBCDIC-to-ASCII translation. The *EOF is coded In position 1 through 4 of the translate table source.

***TRAIL statement** - This statement specifies the trailing character to be stripped in a EBCDIC-to-ASCII translate, and the default pad character for a ASCII-to-EBCDIC translate. *TRAIL is coded in positions 1 through 6 and the trailing character (in hex) Is coded In position 19 and 20. No other data except comments is valid in this statement.

***1 - *9** - These entries define additional End-of-Record and file "TO HEX" characters. This statement is coded with the * in position 1 and the number 1 through 9 in position 2. The remainder of the "FROM HEX" field (positions 3 through 18) must be blank. The only valid entries on this statement are the "TO HEX" characters and comments.

Comment statements - Comment statements can be included in the translate table source by coding starting in position 1.

Sample Translate Table

Table 4-4 illustrates an example of a character-oriented ASCII-to-EBCDIC translation table. For a complete listing, see source member A@E in the library member provided on the S/36 distribution diskette. It can be printed with the Source Entry Utility or the SSP procedure LISTLIBR.

Table 4-4 * Sample ASCII-to-EBCDIC Translation Table

FROM HEX	TO HEX	XBA	PAD	EOR	EOF	Comments
01						Omit unprintable
02						characters
.						
.						
30	FO					Numbers
31	F1					
32	F2					
33	F3					
34	F4					
.						
.						
41	C1					Alphabetical
42	C2					
43	C3					
.						
.						
0D0A		B		Y		CR/LF: End-of-record
1A		B			Y	End of file

Field-Oriented Translation

Field-oriented translation tables are stored in a library "object" member. These members can be edited by using the EDITABLE procedure. Unlike the character-oriented source member, these translation tables do not need to be compiled once they have been edited.

A source member describing the field layouts of the data file is required when using the field-oriented translation method. Standard RPG F and I specifications are used in describing the file, making it possible to automatically support all field types including packed and binary. Note that no record identification codes are allowed in these specifications; therefore only one record type may be in the file. The order in which the fields are described in the “I” specifications is the order in which they will appear in the translated file. This makes it possible to rearrange output fields when translating files.

An example of F and I specifications Is shown in Table 4-5.

The columns used in the F and I specifications are listed in Tables 4-6 and 4-7.

Table 4-5 * Sample File & In put Specification Definition

```
Library: USER1       Member: DATA1  S  Statement 1 of 3   Columns  1-80

      0001   FBINARY        P          43       DISK              UPPER
      0002   IBINARY        NS   01
      0003   I                                              1   6 OCUSTNO
      0004   I                                              7   17 NAME
      0005   I                                             18   30 ADDR
      0006   I                                             31   36 CITY
      0007   I                                             37   38 STATE
      0008   I                                             39   43OZIP
      ----------------------END OF LIBRARY MEMBER---------------------
```

Table 4-6 * Valid Entries in the RPC F Field Definition Specification

Columns	**Description**
6	FORM TYPE must contain an “I”
7	COMMENT (optional) may contain an to make this a documentation record.
7-14	FILE NAME (optional) may Contain a valid file name for documentation purposes.
24-27	RECORD LENGTH contains the record length of the file.
29-30	KEY LENGTH (optional) contains the length of the file's key if it is an indexed file.
31	KEY FORMAT (optional) is the format of the file's key if this is an indexed file. “A” designates an alphabetic key, "P" designates a packed key.
35-38	KEY POSITION (optional) is the starting location in the record of file's key if it is an indexed file.
* All other positions in this record should be left blank (unless this is a comment record).	

Table 4-7 * Valid Entries in the RPC I Field Definition Specification

Columns	Description
6	**FORM TYPE** must contain an "I".
7	**COMMENT** (optional) may contain an "" to make this a documentation record.
14-Jul	**FILE NAME** (optional) may contain a valid file name for documentation purposes.
43	**FIELD TYPE** (optional) specifies the format of the field defined in this record. Blank designates an alphabetic or zone-decimal field. "P" designates a packed field, and 'SB" designates a binary field. Consult the RPG manual for details on the field formats.
44-47	**FROM POSITION** is the starting location of the field in the record.
48-5 1	**TO POSITION** is the ending location of the field in the record.
52	**DECIMAL PLACES** (optional) is required for all numeric fields. It specifies the number of decimal places (from 0 to 9) in the field. Leave this entry blank for alphabetic fields.
53-58	**FIELD NAME** (optional) may document the name of the field.
60-80	**FIELD DESCRIPTION** (optional) may document the description of the field. This is an extension to the IBM specification and is not normally allowed. Since the data normally contained in these columns is not used, this allows for better documentation of the fields used.
* All other positions in the record should be left blank (unless this is a comment record).	

Sample Translate Table

Table 4-8 illustrates an example of a field-oriented EBCDIC-to-ASCII translation table. This is the same table provided in the EDITABLE function when a default EBCDIC-to-ASCII table is requested.

Table 4-8 * Sample Field-Oriented Translation Table

Translate Table Name: SAMPLE

Table Description: Sample EBCDIC-TO-ASCII Translation Table

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
1	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
2	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
3	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
4	20	20	20	20	20	20	20	20	20	20	20	2E	3C	28	20	20
5	26	20	20	20	20	20	20	20	20	20	21	24	2A	29	3B	20
6	2D	2F	20	20	20	20	20	20	20	20	7C	2C	25	SF	3E	3F
7	20	20	20	20	20	20	20	20	20	60	3A	23	40	27	3D	22
8	20	61	62	63	64	65	66	67	68	69	20	20	20	20	20	20
9	20	6A	6B	6C	60	6E	6F	70	71	72	20	20	20	20	20	20
A	20	7E	73	74	75	76	77	78	79	7A	20	20	20	20	20	20
B	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
C	7B	41	42	43	44	45	46	47	48	49	20	20	20	20	20	20
D	70	4A	4B	4C	4D	4E	4F	50	51	52	20	20	20	20	20	20
E	5C	20	53	54	55	56	57	58	59	5A	20	20	20	20	20	20
F	30	31	32	33	34	35	36	37	38	39	20	20	20	20	20	20

Transfer/Translate Example

Table 4-9 shows a sample Systern/36 procedure that can be executed by a remote PC user while signed on to the host. The procedure illustrates how PC data, first uploaded to the host for processing, can be transferred back to the PC as a spreadsheet file and a report file. The report can then be printed on the slow PC printer after the PC is off-line with the host, thus saving long-distance line charges. The nine program steps are explained below the table.

Table 4-9

```
1   // * 'DAILY SUMMARY PROCESSING'
2.  // IF DATAF1-DAILY DELETE DAILY, F1
3.  ETU36 FILEFRPC,A:\PCDAILY.CUR,DAILY 128,T,*BASICS, ,2000,TRANSPEC,USERLIB
4.  PROCESS DAILY, 05, 5000,NO
5. ETU36 DELETE, C:\SALES\PCDAILY.PRV,,,NO
6.  ETU36 TESTFILE, C:\SALES\PCDAIL.TRN ,,,1
7.  // IF ?L'1,4?/0000 ETU36 RENAME,,,C:\SALES\PCDAILY.PRV
8. ETU36 FILETOPC, SUMMARY,,*BASICS,,1500,SUMMSPEC,USERLIB,
    C:\SALES\SUMMARY.PRT
9. ETU36 PRNTTOPC, COPYPRT, FSUMM, CANCEL, 1000,1.132,C:\SALES\SUMMARY.PRT
```

Program steps:

1. Message to the PC operator that the procedure is executing.

2. Precautionary DELETE of System/36 file DAILY, if It exists.

3. Transmit the diskette file A:\PCDAILY.CUR from the PC to the System/36 disk under the name DAILY, with field-oriented translation. The field definition member TRANSPEC is contained in library USERLIB.

4. System/36 user-written procedure.

S. Delete the PC file C:\SALES\PCDAILY.PRV and do not halt if it does not exist.

6. Test the PC for file C:\SALES\PCDAILY.TRN and place the return code and number of bytes in LDA position 1-12.

7. Check the return code from the previous TESTFILE function and if the file exists, then RENAME the file to C:\SALES\PCDAILY.PRV.

8. Transfer the System/36 file SUMMARY to the PC under the name C:\SALES\SUMMARY.PRT and translate the file into the BASIC Sequential format so it can be easily imported into a spreadsheet such as LOTUS 1-2-3.

9. Transfer the report on the System/36 print spool with the forms type of "SUMM" to the PC under the name C:\SALES\:SUMMARY.PRT for later printing while offline from the System/36

Appendix A

ASCII/EBCDIC PRINTABLE CHARACTERS

Dec	Hex	EBCDIC	ASCII
0	00		
1	01		
2	02		
3	03		
4	04		
5	05		
6	06		
7	07		
8	08		
9	09		
10	0A		
11	0B		
12	0C		
13	0D		
14	0E		
15	0F		
16	10		
17	11		
18	12		
19	13		
20	14		
21	15		
22	16		
23	17		
24	18		
25	19		
26	1A		
27	1B		
28	1C		
29	1D		
30	1E		
31	1F		
32	20		space
33	21		
34	22		"
35	23		#
36	24		$
37	25		%
38	26		&
Dec	**Hex**	**EBCDIC**	**ASCII**

39	27		'
40	28		(
41	29		)
42	2A		*
43	2B		+
44	20		'
45	2D		
46	2E		.
47	2F		/
48	30		0
49	31		1
50	32		2
51	33		3
52	34		4
53	35		5
54	36		6
55	37		7
56	38		8
57	39		9
58	3A		:
59	3B		;
60	30		
61	3D		=
62	3E		
63	3F		?
64	40	space	@
65	41		A
55	42		B
57	43		C
58	44		D
69	45		B
70	45		F
71	47		G
72	48		H
73	49		I
74	4A	cents	J
75	4B	.	K
76	40		L
77	4D	(	M
78	4E	+	N
79	4F	or	O
80	50	&	P
81	51		Q
82	52		R
83	53		S
84	54		T
85	55		U
Dec	**Hex**	**EBCDIC**	**ASCII**

Dec	Hex	EBCDIC	ASCII
86	56		V
87	57		W
88	58		X
89	59		Y
90	5A	**!**	Z
91	5B	$	[
92	5C	*	/
93	5D	)	]
94	5E	**;**	^
95	5F	^	
96	60		
97	61	/	a
98	52		b
99	63		c
100	64		d
101	65		e
102	66		f
103	67		g
104	68		h
105	69		I
106	6A	**!**	j
107	6B	**,**	k
108	6C	%	l
109	6D		m
110	6E		n
111	6F	?	o
112	70		p
113	71		q
114	72		r
115	73		s
116	74		t
117	75		u
118	76		v
119	77		w
120	78		x
121	79		y
122	7A	**:**	z
123	7B	#	**{**
124	70	@	**!**
125	7D	'	**}**
126	7B	=	**"**
127	7F	"	
128	80		
129	81	a	
130	82	b	
131	83	c	
132	84	d	

133	85	e	
134	86	f	
135	87	g	
136	88	h	
137	89	I	
138	8A		
139	8B		
140	80		
141	8D		
142	8E		
143	8F		
144	90		
145	91	j	
146	92	k	
147	93	1	
148	94	m	
149	95	n	
150	96	o	
151	97	p	
152	98	q	
153	99	r	
154	9A		
155	9B		
156	9C		
157	9D		
157	9E		
158	9F		
160	A0		
161	A1		
162	A2	s	
163	A3	t	
164	A4	u	
165	A5	v	
166	A6	w	
167	A7	x	
168	A8	y	
169	A9	z	
170	AA		
171	AB		
172	AC		
173	AD		
174	AE		
175	AF		
176	B0		
177	B1		
178	B2		
179	B3		
Dec	**Hex**	**EBCDIC**	**ASCII**

180	B4		
181	B5		
182	B6		
183	B7		
184	B8		
185	B9		
186	BA		
187	BB		
188	BC		
189	BD		
190	BE		
191	BF		
192	C0	{	
193	C1	A	
194	C2	B	
195	C3	C	
196	C4	D	
197	C5	E	
198	C6	F	
199	C7	G	
200	C8	H	
201	C9	I	
202	CA		
203	CB		
204	CC		
205	CD		
206	CE		
207	CF		
208	D0	}	
209	D1	J	
210	D2	K	
211	D3	L	
212	D4	M	
213	DS	N	
214	D6	0	
215	D7	P	
216	D8	Q	
217	D9	R	
218	DA		
219	DB		
220	DC		
221	DD		
222	DE		
223	DF		
224	E0		
225	El		
226	E2	S	
Dec	**Hex**	**EBCDIC**	**ASCII**

227	E3	T
228	E4	U
229	ES	V
230	E6	W
231	E7	X
232	E8	Y
233	E9	Z
234	EA	
235	EB	
236	EC	
237	ED	
238	EE	
239	EF	
240	F0	0
241	Fl	1
242	F2	2
243	F3	3
244	F4	4
245	F5	5
246	F6	6
247	F7	7
248	F8	8
249	F9	9
250	FA	
251	FB	
252	FC	
253	FD	
254	FE	
255	FF	

Appendix B

International Translation Tables

Installing International Translation Tables

Ringdale supports sixteen languages for its ETU/36 file transfer programs on System/36 and AS/36 host computers:

Country	ID
Belgian French	BF
Canadian French	CF
Danish	DK
French	FR
German	GR
Italian	IT
Dutch	NL
Norwegian	NO
Portuguese	P0
Swiss French	SF
Swiss German	SG
Spanish	SP
Finnish	SU
Swedish	SV
UK English	UK
US English	US (default)

Refer to the section below if you need to install non-US translation table on your host machine. If you are using the US English table, you do not need to install the files on this diskette.

The System/36 and Advance System 36 Host

Insert the diskette in the host diskette drive. Enter these commands:

```
TOLIBR SRCxx, , , ,PCTRAN      [where xx is the country ID]
// LIBRARY NAME-PCTRAN
CRTTAB xx                  [where xx is the country ID]
```

Appendix C

MESSAGES

ETU/36 Messages

The following are the message numbers and their associated text from the ETU. The System/36 supports up to four possible response options to a message. The ETU will allow the appropriate response options depending upon circumstances.

Table B-1

Option	Meaning and Resulting Action
0	Informational message. Program continues if this option is selected.
1	Recoverable error. Selecting this option will retry the operation.
2	Terminal error. The ETU function is terminated and the System/36 procedure
3	Terminal error. The ETU function is terminated and the job is cancelled.

Message Description

*** S/36 Emulation Transfer Message Member**

0001 PCTRAN - FUNCTION PARAMETER IS INVALID -JOB IS CANCELLED
0002 PCTRAN - INVALID OR MISSING PARAMETER -JOB IS CANCELLED
0015 PCTPAN - FILE TRANSLATION PROCEDURE EXECUTING
0016 PCTRAN - TRANSLATE TABLE COMPILER PROCEDURE EXECUTING
0017 PCTRAN - FILE TRANSLATION PROCEDURE EXECUTING
0019 PCTRAN - FUNCTION CANNOT BE RUN FROM INQUIRY MODE
0020 PCTPAN - FUNCTION MUST BE RUN WITH A REQUESTOR -JOB IS CANCELLED
0021 PCTRAN - FUNCTION CANNOT BE RUN FROM INQUIRY MOD -JOB IS CANCELLED
0030 PCTRAN - INVALID OR MISSING PARAMETER FOR TRANSLATE TABLE COMPILER
0041 PCTRAN - ERROR IN TRANSLATE TABLE FILE
0042 PCTRAN - ERROR IN TRANSLATE TABLE FILE
0043 PCTRAN - ERROR IN TRANSLATE TABLE FILE
0044 PCTRAN - ERROR IN TRANSLATE TABLE FILE
0045 PCTPAN - ERROR IN TRANSLATE TABLE FILE
0046 PCTRAN - ERROR IN TRANSLATE TABLE FILE
0047 PCTRAN - ERROR IN TRANSLATE TABLE FILE
0048 PCTRAN - ERROR IN TRANSLATE TABLE FILE

0049 PCTRAN - ERROR IN TRANSLATE TABLE FILE
0050 PCTRAN - ERROR IN TRANSLATE TABLE FILE
0051 PCTRAN - ERROR IN TRANSLATE TABLE FILE
0052 PCTRAN - ERROR IN TRANSLATE TABLE FILE
0053 PCTRAN - MISSING OR INVALID PARAMETER FOR TRANSLATE
0054 PCTRAN - ERROR IN TRANSLATE TABLE FILE (UNEXPECTED END OF FILE)
0055 PCTRAN - DISK ERROR ON OUTPUT FILE
0056 PCTPAN - OUTPUT FILE FULL
0057 PCTRAN - INVALID RECORD LENGTH - MUST BE 1-4096
0058 PCTRAN - FILE DOES NOT EXIST

*** Messages for Program PC#60: tran table edit program**

0601 OCL ERROR - PREMATURE END OF OCL OPTIONS - MISSING MEMBER OR TYPE
0602 OCL ERROR - OCL INPUT LINES IN INCORRECT SEQUENCE
0603 OCL ERROR - MODULE OR LIBRARY NAME BLANK OR INVALID
0604 OCL ERROR - TYPE IS NOT CREATE OR UPDATE
0605 OCL ERROR - MODULE SPECIFIED HAS INCORRECT SIZE FOR TRANSLATE TABLE
0606 OCL ERROR - MODULE SPECIFIED HAS INCORRECT SIZE FOR TRANSLATE TABLE
0607 OCL ERROR - CREATE SPECIFIED, BUT MODULE ALREADY EXISTS
0608 I/O ERROR - WRITE ERROR WHILE TRYING TO WRITE TO DISPLAY
0609 I/O ERROR - WRITE ERROR WHILE TRYING TO WRITE RESULTS TO OUTPUT FILE
0610 OCL ERROR - TYPE OF DEFAULT TABLE IS NOT 1, 2, 3, OR 4
0611 OCL ERROR - DEFAULT TABLE TO COPY NOT FOUND IN LIBRARY
0612 OCL ERROR - MODULE SPECIFIED HAS INCORRECT SIZE FOR TRANSLATE TABLE
0621 PROGRAM ERROR - LOGIC ERROR FOUND WHILE PROCESSING
1000 P.C. FILE NOT FOUND IN DIRECTORY
1001 P.C. HARDWARE ERROR - MEDIA IS WRITE PROTECTED
1002 P.C. DRIVE IS CURRENTLY UNAVAILABLE
1003 P.C. HARDWARE ERROR - DISK NOT READY -
1004 P.C. HARDWARE ERROR - UNKNOWN COMMAND RECEIVED
1005 P.C. HARDWARE ERROR - INVALID DATA READ
1006 P.C. HARDWARE ERROR - BAD REQUEST STRUCTURE LENGTH
1007 P.C. HARDWARE ERROR - BAD SEEK
1008 P.C. HARDWARE ERROR - UNKNOWN MEDIA TYPE
1009 P.C. HARDWARE ERROR - SECTOR NOT FOUND
1010 P.C. HARDWARE ERROR - PRINTER OUT OF PAPER
1011 P.C. HARDWARE ERROR - WRITE FAULT
1012 P.C. HARDWARE ERROR - READ FAULT
1013 P.C. HARDWARE ERROR - STATUS UNKNOWN

* **Messages for Program PC#1 1:**
*** REFORMAT OF FILE INTO PC ASCII/DIF FORMAT**

1100 OCL ERROR - INPUT FILE OCL MISSING OR INVALID
1101 OCL ERROR - INPUT FILE OCL MUST ATF~ACH TO EXISTING DATA FILE
1102 OCL ERROR - TRANSLATE TABLE SPECIFIED NOT FOUND IN LIBRARY AT RUNTIME

1103 DATA ERROR - INPUT FILE TO BE REFORMATTED HAS NO DATA RECORDS
1104 OCL ERROR - TRANSLATE TABLE SPECIFIED IS NOT THE CORRECT LENGTH
1105 OCL ERROR - TRANSLATE TABLE OCL STATEMENT MISSING OR INVALID
1110 I/OERROR - SYSTEM INPUT ERROR ON FILE $SOURCE : INPUT SPECS OF FILE
1111 I/O ERROR - SYSTEM INPUT ERROR ON USER INPUT FILE TO BE REFORMATTED
1112 I/O ERROR - SYSTEM OUTPUT ERROR ON REFORMATTED DATA FILE
1113 DATA ERROR - NO FILE INPUT SPECIFICATIONS FOUND IN LIBRARY MEMBER
1114 DATA ERROR - NON "INPUT SPECIFICATION" FOUND IN FILE INPUT SPECS
1115 DATA ERROR- INPUT SPECS: FILE IDENTIFIER RECORD MISSING OR INVALID
1116 DATA ERROR - INPUT SPECS: INVALID ENTRY IN FIELD TYPE CODE
1117 DATA ERROR - INPUT SPECS: PACKED OR BINARY FIELD MISSING DECIMAL POS.
1118 DATA ERROR - INPUT SPECS: # OF DECIMAL POS. INVALID FOR FIELD LENGTH
1119 DATA ERROR - INPUT SPECS: DECIMAL POS. ENTRY NOT VALID FOR TYPE CODE
1120 DATA ERROR - INPUT SPECS: FIELD LENGTH TOO LARGE FOR DATA 'IYPE
1121 PROGRAM ERROR - LOGIC ERROR FOUND IN PROCESSING
1122 DATA ERROR - INPUT SPECS: INVALID BINARY FIELD LENGTH (NOT 2 OR 4)
1123 DATA ERROR - INPUT SPECS: INVALID VALUE IN "FROM" FIELD
1124 DATA ERROR - INPUT SPECS INVALID VALUE IN "TO" FIELD
1125 DATA ERROR - INPUT SPECS: VALUE IN "FROM" FIELD CANNOT BE ZERO
1126 DATA ERROR - INPUT SPECS: 'FROM" VALUE CANNOT BE GREATER THAN "TO"
1127 DATA ERROR - INPUT SPECS: "TO" VALUE CANNOT BE GREATER THAN 512
1128 DATA ERROR - INPUT SPECS: "TO" VALUE CANNOT BE GREATER THAN RECORD LENGTH
1129 DATA ERROR - INPUT FILE LENGTH IS OVER MAXIMUM FOR PROGRAM (512 BYTES)
1130 DATA ERROR - TOO MANY INPUT SPECIFICATIONS FOR THIS PROGRAM
1131 DATA ERROR - A LOW HEXADECIMAL VALUE (BELOW HEX 40) FOUND IN DATA FIELD

* **PCTRAN Message**

1200 PCTRAN - PROGRAM ERROR

*** Messages for Program PC#12:**
*** REFORMAT OF FILE INTO PC ASCII/BASIC FORMAT**

1202 OCL ERROR - TRANSLATE TABLE SPECIFIED NOT FOUND IN LIBRARY AT RUNTIME
1204 OCL ERROR - TRANSLATE TABLE SPECIFIED IS NOT THE CORRECT LENGTH
1205 OCL ERROR - TRANSLATE TABLE OCL STATEMENT MISSING OR INVALID
1210 I/O ERROR - SYSTEM INPUT ERROR ON FILE $SOURCE: INPUT SPECS OF FILE
1211 I/O ERROR - SYSTEM INPUT ERROR ON USER INPUT FILE TO BE REFORMATTED

1212 I/O ERROR - SYSTEM OUTPUT ERROR ON REFORMA'ITED DATA FILE
1213 DATA ERROR - NO FILE INPUT SPECIFICATIONS FOUND IN LIBRARY MEMBER
1214 DATA ERROR - NON "INPUT SPECIFICATION" FOUND IN FILE INPUT SPECS
1215 DATA ERROR - INPUT SPECS: FILE IDENTIFIER RECORD MISSING OR INVALID
1216 DATA ERROR - INPUT SPECS: INVALID ENTRY IN FIELD TYPE CODE
1217 DATA ERROR - INPUT SPECS: PACKED OR BINARY FIELD MISSING DECIMAL POS.
1218 DATA ERROR - INPUT SPECS: # OF DECIMAL POS. INVALID FOR FIELD LENGTH
1219 DATA ERROR - INPUT SPECS: DECIMAL POS. ENTRY NOT VALID FOR TYPE CODE
1220 DATA ERROR - INPUT SPECS: FIELD LENGTH TOO LARGE FOR DATA TYPE
1221 PROGRAM ERROR - LOGIC ERROR FOUND IN PROCESSING
1222 DATA ERROR - INPUT SPECS: INVALID BINARY FIELD LENGTH (NOT 2 OR 4)
1223 DATA ERROR - INPUT SPECS: INVALID VALUE IN "FROM" FIELD
1224 DATA ERROR - INPUT SPECS: INVALID VALUE IN "TO" FIELD
1225 DATA ERROR - INPUT SPECS: VALUE IN "FROM" FIELD CANNOT BE ZERO
1226 DATA ERROR - INPUT SPECS: "FROM" VALUE CANNOT BE GREATER THAN "TO"
1227 DATA ERROR - INPUT SPECS: "TO" VALUE CANNOT BE GREATER THAN 512
1228 DATA ERROR - INPUT SPECS: "TO" VALUE CANNOT BE GREATER THAN RECORD LENGTH
1229 DATA ERROR - INPUT FILE LENGTH IS OVER MAXIMUM FOR PROGRAM (512 BYTES)
1230 DATA ERROR - TOO MANY INPUT SPECIFICATIONS FOR THIS PROGRAM
1231 DATA ERROR - A LOW HEXADECIMAL VALUE (BELOW HEX 40) FOUND IN DATA FIELD

*** PCTRAN Messages**

2000 P.C. DIRECTORY IS FULL OR INVALID FILE NAME
2001 P.C. HARDWARE ERROR - MEDIA IS WRITE PROTECTED
2002 P.C. DRWE IS CURRENTLY UNAVAILABLE
2003 P.C. HARDWARE ERROR - DISK NOT READY
2004 P.C. HARDWARE ERROR - UNKNOWN COMMAND RECEIVED
2005 P.C. HARDWARE ERROR - INVALID DATA READ
2006 P.C. HARDWARE ERROR - BAD REQUEST STRUCTURE LENGTH
2007 P.C. HARDWARE ERROR - BAD SEEK
2008 P.C. HARDWARE ERROR - UNKNOWN MEDIA TYPE
2009 P.C. HARDWARE ERROR - SECTOR NOT FOUND
2010 P.C. HARDWARE ERROR - PRINTER OUT OF PAPER
2011 P.C. HARDWARE ERROR - WRITE FAULT
2012 P.C. HARDWARE ERROR - READ FAULT
2013 P.C. HARDWARE ERROR- STATUS UNKNOWN
2100 P.C. DISK IS FULL
2200 PCTRAN - PROGRAM ERROR
3000 P.C. DIRECTORY IS FULL OR INVALID FILE NAME
3001 P.C. HARDWARE ERROR - MEDIA IS WRITE PROTECTED
3002 P.C. DRIVE IS CURRENTLY UNAVAILABLE
3003 P.C. HARDWARE ERROR - DISK NOT READY

3004 P.C. HARDWARE ERROR - UNKNOWN COMMAND RECEIVED
3005 P.C. HARDWARE ERROR - INVALID DATA READ
3006 P.C. HARDWARE ERROR - BAD REQUEST STRUCTURE LENGTH
3007 P.C. HARDWARE ERROR - BAD SEEK
3008 P.C. HARDWARE ERROR - UNKNOWN MEDIA TYPE
3009 P.C. HARDWARE ERROR - SECTOR NOT FOUND
3010 P.C. HARDWARE ERROR - PRINTER OUT OF PAPER
3011 P.C. HARDWARE ERROR - WRITE FAULT
3012 P.C. HARDWARE ERROR - READ FAULT
3013 P.C. HARDWARE ERROR - STATUS UNKNOWN
4000 P.C. FILE NOT FOUND IN DIRECTORY
4001 P.C. HARDWARE ERROR - MEDIA IS WRITE PROTECTED
4002 P.C. DRIVE IS CURRENTLY UNAVAILABLE
4003 P.C. HARDWARE ERROR - DISK NOT READY
4004 P.C. HARDWARE ERROR - UNKNOWN COMMAND RECEIVED
4005 P.C. HARDWARE ERROR - INVALID DATA READ
4006 P.C. HARDWARE ERROR - BAD REQUEST STRUCTURE LENGTH
4007 P.C. HARDWARE ERROR - BAD SEEK
4008 P.C. HARDWARE ERROR - UNKNOWN MEDIA TYPE
4009 P.C. HARDWARE ERROR - SECTOR NOT FOUND
4010 P.C. HARDWARE ERROR - PRINTER OUT OF PAPER
4011 P.C. HARDWARE ERROR - WRITE FAULT
4012 P.C. HARDWARE ERROR - READ FAULT
4013 P.C. HARDWARE ERROR - STATUS UNKNOWN
5000 P.C. FILE NOT FOUND IN DIRECTORY
5001 P.C. HARDWARE ERROR - MEDIA IS WRITE PROTECTED
5002 P.C. DRIVE IS CURRENTLY UNAVAILABLE
5003 P.C. HARDWARE ERROR - DISK NOT READY
5004 P.C. HARDWARE ERROR - UNKNOWN COMMAND RECEIVED
5005 P.C. HARDWARE ERROR - INVALID DATA READ
5006 P.C. HARDWARE ERROR - BAD REQUEST STRUCTURE LENGTH
5007 P.C. HARDWARE ERROR - BAD SEEK
5008 P.C. HARDWARE ERROR - UNKNOWN MEDIA TYPE
5009 P.C. HARDWARE ERROR - SECTOR NOT FOUND
5010 P.C. HARDWARE ERROR - PRINTER OUT OF PAPER
5011 P.C. HARDWARE ERROR - WRITE FAULT
5012 P.C. HARDWARE ERROR - READ FAULT
5013 P.C. HARDWARE ERROR - STATUS UNKNOWN
6000 P.C. FILE NOT FOUND OR NEW NAME ALREADY EXISTS IN DIRECTORY
6001 P.C. HARDWARE ERROR - MEDIA IS WRITE PROTECTED
6002 P.C. DRIVE IS CURRENTLY UNAVAILABLE
6003 P.C. HARDWARE ERROR - DISK NOT READY
6004 P.C. HARDWARE ERROR - UNKNOWN COMMAND RECEIVED
6005 P.C. HARDWARE ERROR - INVALID DATA READ
6006 P.C. HARDWARE ERROR - BAD REQUEST STRUCTURE LENGTH
6007 P.C. HARDWARE ERROR - BAD SEEK
6008 P.C. HARDWARE ERROR - UNKNOWN MEDIA TYPE
6009 P.C. HARDWARE ERROR - SECTOR NOT FOUND
6010 P.C. HARDWARE ERROR - PRINTER OUT OF PAPER
6011 P.C. HARDWARE ERROR - WRITE FAULT
6012 P.C. HARDWARE ERROR - READ FAULT
6013 P.C. HARDWARE ERROR - STATUS UNKNOWN

*** Messages for Program PC#71:**
*** REFORMAT OF FILE from PC ASCII/DIF FORMAT**

7102 OCL ERROR - TRANSLATE TABLE SPECIFIED NOT FOUND IN LIBRARY AT RUNTIME
7104 OCL ERROR - TRANSLATE TABLE SPECIFIED IS NOT THE CORRECT LENGTH
7105 OCL ERROR - TRANSLATE TABLE OCL STATEMENT MISSING OR INVALID
7110 I/O ERROR - SYSTEM INPUT ERROR ON FILE $SOURCE: INPUT SPECS OF FILE
7111 I/O ERROR - SYSTEM INPUT ERROR ON USER INPUT FILE TO BE REFORMATTED
7112 I/O ERROR - SYSTEM OUTPUT ERROR ON REFORMATITED DATA FILE
7113 DATA ERROR - NO FILE INPUT SPECIFICATIONS FOUND IN LIBRARY MEMBER
7114 DATA ERROR - NON "INPUT SPECIFICATION" FOUND IN FILE INPUT SPECS
7115 DATA ERROR - INPUT SPECS: FILE IDENTIFIER RECORD MISSING OR INVALID
7116 DATA ERROR - INPUT SPECS: INVALID ENTRY IN FIELD TYPE CODE
7117 DATA ERROR - INPUT SPECS: PACKED OR BINARY FIELD MISSING DECIMAL POS.
7118 DATA ERROR - INPUT SPECS: # OF DECIMAL POS. INVALID FOR FIELD LENGTH
7119 DATA ERROR - INPUT SPECS: DECIMAL POS. ENTRY NOT VALID FOR TYPE CODE
7120 DATA ERROR - INPUT SPECS: FIELD LENGTH TOO LARGE FOR DATA TYPE
7121 PROGRAM ERROR - LOGIC ERROR FOUND IN PROCESSING
7122 DATA ERROR - INPUT SPECS: INVALID BINARY FIELD LENGTH (NOT 2 OR 4)
7123 DATA ERROR - INPUT SPECS: INVALID VALUE IN "FROM' FIELD
7124 DATA ERROR - INPUT SPECS: INVALID VALUE IN "TO' FIELD
7125 DATA ERROR - INPUT SPECS: VALUE IN "FROM" FIELD CANNOT BE ZERO
7126 DATA ERROR - INPUT SPECS: "FROM" VALUE CANNOT BE GREATER THAN 'TO"
7127 DATA ERROR - INPUT SPECS: "TO" VALUE CANNOT BE GREATER THAN 512
7128 DATA ERROR - INPUT SPECS: "TO" VALUE CANNOT BE GREATER THAN RECORD LENGTH
7129 DATA ERROR- INPUT FILE LENGTH IS OVER MAXIMUM FOR PROGRAM (512 BYTES)
7130 DATA ERROR - TOO MANY INPUT SPECIFICATIONS FOR THIS PROGRAM
7131 DATA ERROR - UNEXPECTED OR INCORRECT DATA IN THE DIF HEADER INFORMATION
7132 DATA ERROR - UNKNOWN TYPE OF DIP "SPECIAL DATA VALUE" ... (-1,0) ITEM
7133 DATA ERROR - TOO MANY FIELDS IN ASCII DATA FOR INPUT SPECIFICATIONS
7134 DATA ERROR - DATA MISMATCH BETWEEN ASCII DATA AND INPUT SPECS FIELD TYPE
7135 DATA ERROR - ASCII NUMERIC VALUE TO CONVERT NOT FOUND
7136 DATA ERROR - BAD CHARACTER FOUND IN ASCII NUMERIC VALUE
7137 DATA ERROR - UNEXPECTED END-OF-FILE CEOF) ENCOUNTERED IN ASCII FILE
7138 DATA ERROR - ASCII VARIABLE LENGTH INPUT RECORD OVER 1024 BYTES

7139 DATA ERROR - BAD OUTPUT FILE RECORD LENGTH IN INPUT SPECIFICATIONS
7140 DATA ERROR - OUTPUT FILE RECORD LENGTH IN INPUT SPECIFICATION OVER 512
7141 DATA ERROR - LOGICAL ASCII INPUT RECORD OVER 258 BYTES
7142 DATA ERROR - ERROR IN OUTPUT FILE KEY LENGTh SPECIFIED IN INPUT SPECS
7143 DATA ERROR - ERROR IN OUTPUT FILE KEY DISP SPECIFIED IN INPUT SPECS
7144 DATA ERROR - NUMERIC VALUE IN FILE IN E" FORMAT WITH EXPONENT TOO LARGE

*** Messages for Program PC#72: ***
*** REFORMAT OF FILE from PC ASCII/BASIC FORMAT**

7202 OCL ERROR - TRANSLATE TABLE SPECIFIED NOT FOUND IN LIBRARY AT RUNTIME
7204 OCL ERROR - TRANSLATE TABLE SPECIFIED IS NOT THE CORRECT LENGTH
7205 OCL ERROR - TRANSLATE TABLE OCL STATEMENT MISSING OR INVALID
7210 I/O ERROR - SYSTEM INPUT ERROR ON FILE $SOURCE : INPUT SPECS OF FILE
7211 I/O ERROR - SYSTEM INPUT ERROR ON USER INPUT FILE TO BE REFORMX1TED
7212 I/O ERROR - SYSTEM OUTPUT ERROR ON REFORMATTED DATA FILE
7213 DATA ERROR - NO FILE INPUT SPECIFICATIONS FOUND IN LIBRARY MEMBER
7214 DATA ERROR - NON 'INPUT SPECIFICATION" FOUND IN FILE INPUT SPECS
7215 DATA ERROR - INPUT SPECS: FILE IDENTIFIER RECORD MISSING OR INVALID
7216 DATA ERROR - INPUT SPECS: INVALID ENTRY IN FIELD 'IYPE CODE
7217 DATA ERROR - INPUT SPECS: PACKED OR BINARY FIELD MISSING DECIMAL P05.
7218 DATA ERROR - INPUT SPECS: # OF DECIMAL POS. INVALID FOR FIELD LENGTH
7219 DATA ERROR - INPUT SPECS: DECIMAL POS. ENTRY NOT VALID FOR 'IYPE CODE
7220 DATA ERROR - INPUT SPECS: FIELD LENGTH TOO LARGE FOR DATA TYPE
7221 PROGRAM ERROR - LOGIC ERROR FOUND IN PROCESSING
7222 DATA ERROR - INPUT SPECS: INVALID BINARY FIELD LENGTH (NOT 2 OR 4)
7223 DATA ERROR - INPUT SPECS: INVALID VALUE IN "FROM" FIELD
7224 DATA ERROR - INPUT SPECS: INVALID VALUE IN "TO" FIELD
7225 DATA ERROR - INPUT SPECS: VALUE IN "FROM" FIELD CANNOT BE ZERO
7226 DATA ERROR - INPUT SPECS: "FROM" VALUE CANNOT BE GREATER THAN "TO"
7227 DATA ERROR - INPUT SPECS: "TO" VALUE CANNOT BE GREATER THAN 512
7228 DATA ERROR - INPUT SPECS: "TO" VALUE CANNOT BE GREATER THAN RECORD LENGTH
7229 DATA ERROR - INPUT FILE LENGTH IS OVER MAXIMUM FOR PROGRAM (512 BYTES)
7230 DATA ERROR - TOO MANY INPUT SPECIFICATIONS FOR THIS PROGRAM
7233 DATA ERROR - TOO MANY FIELDS IN ASCII DATA FOR INPUT SPECIFICATIONS

7234 DATA ERROR - DATA MISMATCH BETWEEN FIELD DATA AND INPUT SPECS FIELD TYPE
7235 DATA ERROR - NUMERIC VALUE TO CONVERT NOT FOUND
7236 DATA ERROR - BAD CHARACTER FOUND IN ASCII NUMERIC VALUE
7238 DATA ERROR - ASCII VARIABLE LENGTH INPUT RECORD OVER 1024 BYTES
7239 DATA ERROR - BAD OUTPUT FILE RECORD LENGTH IN INPUT SPECIFICATIONS
7240 DATA ERROR - OUTPUT FILE RECORD LENGTH IN INPUT SPECIFICATION OVER 512
7241 DATA ERROR - LOGICAL ASCII INPUT RECORD OVER 258 BYTES
7242 DATA ERROR - ERROR IN OUTPUT FILE KEY LENGTH SPECIFIED IN INPUT SPECS
7243 DATA ERROR - ERROR IN OUTPUT FILE KEY DISP SPECIFIED IN INPUT SPECS
7244 DATA ERROR - NUMERIC VALUE IN FILE IN "E" FORMAT WITH EXPONENT TOO LARGE

*** PCTRAN Messages**

7901 I/O Error - Error on input of data record to Library Control Program
7902 I/O ERROR - Error on output of data record from Library Control Program
7903 System Error - Bad Copyright detected in program
7904 OCL Error - Bad parameter data for "ADD LIBRARY CONTROL" Program
9000 P.C. FILE CANNOT BE CLOSED - DISKETTE WAS CHANGED
9001 P.C. HARDWARE ERROR- MEDIA IS WRITE PROTECTED
9002 P.C. DRIVE IS CURRENTLY UNAVAILABLE
9003 P.C. HARDWARE ERROR - DISK NOT READY
9004 P.C. HARDWARE ERROR - UNKNOWN COMMAND RECEIVED
9005 P.C. HARDWARE ERROR - INVALID DATA READ
9006 P.C. HARDWARE ERROR - BAD REQUEST STRUCTURE LENGTH
9007 P.C. HARDWARE ERROR - BAD SEEK
9008 P.C. HARDWARE ERROR - UNKNOWN MEDIA TYPE
9009 P.C. HARDWARE ERROR - SECTOR NOT FOUND
9010 P.C. HARDWARE ERROR - PRINTER OUT OF PAPER
9011 P.C. HARDWARE ERROR - WRITE FAULT
9012 P.C. HARDWARE ERROR - READ FAULT
9013 P.C. HARDWARE ERROR - STATUS UNKNOWN
9980 PCTPAN - PROGRAM ERROR
9986 PCTRAN - FILE TRANSFER WAS INTERRUPTED: 1=CONTINUE 3=CANCEL
9987 PCTPAN - DEVICE NOT P.C. OR EMULATOR LOADED INCORRECTLY
9988 XXXXXXXX BYTES IN THIS P.C. FILE
9989 PCTRAN - ATTACHED LOCAL WORKSTATION IS NOT A P.C.
9990 PCTRAN - INVALID PARAMETER
9991 PCTPAN - INVALID FUNCTION PARAMETER
9992 PCTRAN - INVALID OR UNSPECIFIED P.C. DRIVE
9993 P.C. FILE NAME NOT SPECIFIED
9994 P.C. FILE EXTENSION IS INVALID
9995 PCTRAN - SYSTEM/36 FILE OCL STATEMENT MISSING OR INVALID
9996 PCTRAN - SYSTEM/36 FILE RECORD LENGTH INVALID MUST BE 1-4096
9997 PCTRAN - INVALID LOCAL DATA AREA OFFSET MUST BE 1-501
9998 PCTRAN - TOO MANY RECORDS TO TRANSFER MAX IS 8,388,607
9999 PCTRAN - WORKSTATION ATTRIBUTES NOT VALID FOR TRANSFER UTILITY

Appendix D

ETU/36 Problem Guide

This is a guide to common problems with ETU/36. It is based on problems that have occurred and the solutions found for them. Search for the exact text of the message you are receiving.

This guide is separated in the following sections: Install questions, translation questions, transfer problems, translation problems, OCL problems, error messages, Mac problems, and USR messages. It is recommended that you use the search function to look for your specific problem.

The latest version of ETU 36 is 4.108.

INSTALLATION QUESTIONS:

QUESTION: Will ETU/36 work (from SSP Machine M36) with ES/TCP emulation or is a twinax card required?

ANSWER: ETU/36 will work with any of NLynx's current emulation products: ES/TCP, ES/Remote, ES/3XTwin, ES/PCI, ES/Server, TCP/Axcess, and MacMidrange Client. In some version of the ES32 (or the older ES/95) software the ETU support needs to be enabled. You do this by selecting Session, then Properties. Select the API tab and put a check in the Enable ETU box.

QUESTION: They have an Advanced 36. What product do I need for file transfer if they are running SSP?

ANSWER: SSP MACHINE mode, where the AS/400 is booted in this mode using *M36 or if you have the IBM 436 (without an OS/400 license) came preconfigured as 'SSP dominant' which basically hides the underlying OS/400 from the user and the machine 'appears' as though it's running SSP. This mode needs ETU/36. If it says "Enter option or command" you are in SSP mode. MENU PCTRAN is the command to start ETU. If you boot the machine in SSP mode, you need ETU 36. You might take a look at this http://www.nlynx.com/html/tb-etuneed.htm

QUESTION: How do I know what media to order for my midrange host?

ANSWER: There is ETU/36 that is for the old machines that have the old 5 1/4" or 8" diskette drive... and there is ETU for AS/36, which is on 1/4" cassette.

PROBLEM: Trying to install software gets this message: "**SYS-2594 ETUXFER Trying to Copy Privileged Module PCRCB1 not found**".

CAUSE: Customer installed the ETU 36 software but not as a security officer.

SOLUTION: The installation requires some privileges that require the use of the security officer privileges. There is no work around as these capabilities are external to ETU and is intrinsic to the SSP. If a person lost the Security Officer password, the whole drive would have to be reformatted.

TRANSLATION QUESTIONS:

QUESTION: What PC File formats can ETU/36 transfer to?

ANSWER: ETU/36 when transferring a file to a PC can transfer the following types:

*TEXT. This is usually used if transferring documents.

*SAVE used for offline storage or to move executables.

*TAB (tab delimited), *BASICS (basic sequential, which is comma-delimited), and *DIF (differential format), are used to transfer database files. You would need to have (or create) an F & I Spec (Field and Input) to do this.

QUESTION: Will a file with no packed fields come down as ASCII if I specify *TEXT as the type of translation?

ANSWER: Yes, all of these reformat types: *DIF, *BASICS, *TAB, and *TEXT convert from EBCDIC to ASCII.

The reformat types: *SAVE, and *NONE do not.

You could also create a “1 to 1” translation table.

QUESTION: Can I transfer specific records from my System 36 host file using ETU/36?

ANSWER: ETU/36 does not have the capability of record selection. Only ETU 400 does. You would have to do the record selection using QRY first.

QUESTION: If I transfer a packed file and then transfer it back again, will it still be there?

ANSWER: Yes, the file will still be there… on the PC and it will still be on the host, although it may or may not be the same, depending on how you write your F & I Specifications. When the file resides on the host, the records all have the same size. The fields in each record have the same size in all records as well. The Fields widths are defined externally in the F & I specification. When the file is transferred to the PC, the F & I spec is not transferred, but it is used to insert delimiters, such as commas (for Basic Sequential), tabs (for tab delimited), or carriage returns for *DIF (Data Interchange Format). When you transfer the file back, the same F & I spec will usually be used, but it may need to be tweaked to resolve differences in the end of file delimiters.

QUESTION: How do I invoke a procedure to run from a command line?

ANSWER: After entering the command, before putting the Workstation File Name, you can see the command line and you could cut and paste it. You would need to add in the **Workstation File Name**, for example: “ETU36 FILETOPC,HOODICKIE,,*TEXT,,,,,”

QUESTION: Does it delete the file after it transfers it?

ANSWER: Yes and no. Technically, this is not a transfer; it is a ‘convert (to ASCII) and copy function’. If the name of the ‘Workstation File’ already exists on the PC, it will overwrite it. And it will delete the copy of the file that was converted to ASCII on the host.

QUESTION: What effect does the "**Record length for the translation work file**" parameter have when I am doing a text file transfer?

ANSWER: It will not have any effect on your data at all. The effect is only to allow for better performance in buffering the data. If a large file was transferring very slowly, it might make a difference to make the records smaller.

QUESTION: What effect does the "**Record length for the translation work file**" parameter have when I am doing a field-oriented file transfer with an F & I spec?

ANSWER: You must tell the software what size the record is. If your data file is 440 characters, and you need carriage return – line feed at the end, you must set it to 442.

PROBLEM: I am trying to use PRTXTOPC and I require data in a DIF format, but I can only get text data.

CAUSE: A spool file does not have a DDS.

SOLUTION(s):

1) You can find or create a DDS, and then use PWRTO3XB.

2) Transfer down as TEXT using PRTXTOPC and use Excel's import features.

PROBLEM: I am using the Translate function with the tab-delimited translation. The alpha fields are ending up with quotation marks around them. How can I eliminate them?

CAUSE: The type of field that you defined in your F and I spec causes this.

SOLUTION: Most import utilities can eliminate these if you do not want them when you import the transferred file in an application.

PROBLEM: I am doing a FROM3X Host File to PC file. File is in QS36F library and each record is 512 bytes long. The last 12 bytes are all blanks. After the file is transferred to the PC the records are 500 bytes long. How do I prevent losing the trailing blanks during the transfer?

CAUSE: Either the STRIP parameter is set to *YES, causing removal of the first 12 bytes (which are the sequence number and date fields) or the ETU version is old.

SOLUTION: It may be a result of the translate table that you have chosen, or that the translate table was modified. This procedure covers how to check that.

1) MENU ETU36

2) Take option 20. Edit Translation Table

3) You will see the translation tables. You are currently using the E@A table.

If page down to the end of the file, you will see this:

```
Columns . . . :   1  71          Edit                ETU504/TRANSSRC
SEU==>                                                  ETOA1
```

```
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
0270.00 FA             FC                  ²
0271.00 FB             96                  û
0272.00 FC             9A                  Ü
0273    FD            97                  ù
0274.00 FE             A3                  ú
0275.00 FF             20
0276.00 *EOR            0D0A                Insert CR,LF at end-of-record
0277.00 *EOF            1A                  Insert x'1A' at end-of-file
0278.00 *TRAIL          40                  Strip trailing blanks       <--REMOVE THIS LINE
      ****************** End of data ****************************************
```

4) Space over the *TRAIL 40 Strip trailing blanks.

5) Press F3.

6) Change the Change/create member . . . . . N to Y

7) Change the Member . . . . . to PCTX.E@AF

8) Press Enter.

9) Press F3.

10) Use the option 21. Compile Text Translation File, selecting your new translation table.

11) When you do the transfer again change the Translate table file ETOA1 to the new one.

PROBLEM: Getting error **"User-1214 Data Error – Non-Input Specification" found in File Input Specs".**

CAUSE: When you do a field-oriented file transfer, the F & I specs have to be in a source member by themselves. ETU/36 cannot find the appropriate F & I specs (in a program, for example) like DFU can.

SOLUTION: You must have a valid F & I Specification for *TAB, *DIF, or *BASICS file transfers. See Appendix F CREATING F & I SPECIFICATIONS

PROBLEM: Getting error user "**7133 DATA ERROR, TOO MANY FIELDS IN ASCII DATA FOR INPUT SPECIFICATIONS**.

CAUSE: You have too many fields in your ASCII data for your input specifications.

SOLUTION: The F & I specification defines a number of fields; the ASCII file has more fields defined than the F & I Spec does. You must either modify the F & I spec or modify the ASCII file.

PROBLEM: Uploading *DIF files to S36...gets error message: "**7134 DATA MISMATCH BETWEEN ASCII DATA AND INPUT SPECS FIELD TYPE**"

CAUSE: You have a field in your ASCII data that cannot possibly be what your input specification says it is. For example if the field is defined as a numeric and contains alpha characters above the hexadecimal range, it could cause this.

SOLUTION: Carefully review your data with your Field and Input specification.

PROBLEM: Got error message "**7233 DATA ERROR - TOO MANY FIELDS IN ASCII DATA FOR INPUT SPECIFICATIONS**" when trying to FILEFRPC.

CAUSE: F & I spec has a problem. The F & I Specifications need to tell ETU where to put the records are in the data. The F & I spec may be pointing two fields to the same data space.

SOLUTION: Columns may be incorrect. For example, the second field starts in column 36 and ends in column 30. Or the previous column ends at 176, and the next column starts at 167. Look for an * in your F & I Spec.

CAUSE: The problem might be a comma inserted in a currency field. The AS400 does not like that.
For example, $1,414.89 should be transferred as $1414.89.

SOLUTION: Look at the line where ETU stopped to see what is different in that record, or the few before it, from all of the previous records. For example, if the balance in a currency field, first went over $1000, it would put in a comma, upsetting the AS400. In Excel, you might try custom format $00000.00, or you can change the cell format to text.

CAUSE: The problem might be a column that is wider than the F & I Specification has allocated for it. Note that if the first record that had data in field G occurred on line 9, the problem might actually be on anther line. For example, if the F & I spec allowed 12 characters for that column, but record 590 had 16 characters in that column, the first time that data was actually in that column, it would fail because the record would be padded out to the 16th character.

SOLUTION: Look at the line where ETU stopped to see what is different in that record, or the few before it, from all of the previous records. Read the cryptic IBM message. Guess what it means. Then verify the width of all of the columns against the F & I Spec.

CAUSE: The 'S/36 file record length' parameter may be set incorrectly.

SOLUTION: This should be equal to the exact maximum record length if all fields are full. Depending on the line ending, you may need to add one or two to this parameter.

PROBLEM: Got error message "**7235 DATA ERROR - NUMERIC VALUE TO CONVERT NOT FOUND**".

CAUSE: I am trying to select specific fields and to transfer them to a different order. The F & I Specifications need to tell ETU where the records are in the data. They are not intended to allow you to re-order the data.

SOLUTION: You can select certain fields, and skip over the others, but you cannot re-order them. Carefully review your data with your Field and Input specification.

TRANSFER PROBLEMS:

PROBLEM: When I do the **ETU36 FILETOPC**, I get this error message: "**ERROR --- USR5255 --- RETURNED FROM NATIVE CODE PROCESSOR**"

CAUSE: This can be caused by transferring to a file that is opened by another high-level application on the PC. An example of 'High level' programs are Word and Excel. Notepad does not cause the problem.

Transferring to a Workstation File Name with a directory or folder that does not exist can also cause this problem. In other words, you can transfer to C:\TEMP\WHAT.TXT, but you cannot transfer to C:\TEST\WHAT.TXT if the TEST directory or folder does not exist.

PROBLEM: Whenever I try to transfer a file I get an error code "**9989 PCTRAN - ATTACHED LOCAL WORKSTATION IS NOT A P.C.** " And this is a PC. It worked until I upgraded from Windows 3.1 to Win98.

CAUSE: ETU only works with PC's or Mac's that have ETU compatible software. ETU/36 will work with any of NLynx emulation products: ES/TCP, ES/Remote, ES/3XTwin, ES/PCI, ES/Server, TCP/Axcess, and MacMidrange Client. There is also a DOS 6.2-based utility called **IBMTRAN** that worked with DOS-based emulation products. This utility was only supported in the DOS mode and does not work with registry-based 32-bit Operating Systems. IBMTRAN is not supported but if you have a DOS-based utility, you can download it for free.

SOLUTION: Upgrade your PC emulation software to ES32.

In some version of the ES32 (or the older ES/95) software the ETU support needs to be enabled. You do this by selecting **Session**, then **Properties**. Select the **API** tab and put a check in the **Enable ETU** box.

If you have a DOS-based emulation product, then download IBMTRAN from ftp://ftp.nlynx.com/pub/Midrange/EmeraldSeries/ETU/IBMTRAN/ IBMT2201.EXE

PROBLEM: ETU does not see my files.

CAUSE: If you are on an Advanced/36 or an AS/400 in SSP mode, and running the MENU ETU36 format, then you must have the files that you need to transfer in the QS36F library. The reason for this anomaly is that on the System 36, data files are global, in other words, they do not go in any particular library, so ETU/36 was not designed look in libraries. With ETU/36 emulation on AS/400 you cannot change which library that it looks for the file in. It has to be in the QS36F library or it will not find the file, unless someone altered the default S/36EE file library. If that is a possibility, you can check with the command DSPS36 and look at S/36 Environment Variables).

SOLUTION: Use the DSPS36 command and take a 5 on S/36 environment values to verify that the Default files library is still QS36F. If it is, and you wish to use ETU/36 mode, then you must put the files in QS36F.

QUESTION: How do I know what files it will see so that it can transfer them?

ANSWER: You can use the (work with files) **WRKF QS36F** command.

PROBLEM: Whenever I try to transfer a file I get an error code "**9987 PCTRAN - DEVICE IS NOT P.C. OR EMULATOR LOADED INCORRECTLY** " This is a PC.

CAUSE: ETU only works with PC's or Mac's that have ETU compatible software. ETU/36 will work with any of NLynx emulation products: ES/TCP, ES/Remote, ES/3XTwin, ES/PCI, ES/Server, TCP/Axcess, and MacMidrange Client. There is also a DOS-based utility called **IBMTRAN** that worked with DOS-based emulation products. This utility was only supported in the

DOS mode and does not work with registry-based 32-bit Operating Systems. IBMTRAN is not supported but if you have a DOS-based utility, you can download it for free.

SOLUTION: Make certain that your emulator is installed correctly. If you have an NLynx emulator, we can help. Currently supported products are:

- ES3XTwin http://www.nlynx.com/html/tb-es3xtwin.htm
- ES/PCI http://www.nlynx.com/html/tb-espci.htm
- ES/TCP http://www.nlynx.com/html/tb-estcp.htm
- ES/Server http://www.nlynx.com/html/tb-esserver.htm
- ES/Remote http://www.nlynx.com/html/es_remote.htm
- TCP/Axcess http://www.nlynx.com/html/tb-tcpaxcess.htm
- MacMidrange Client http://www.nlynx.com/html/tb-macmidrange.htm

Upgrade your PC emulation software to ES32.

In some version of the ES32, (or the older ES/95) software the ETU support needs to be enabled. You do this by selecting **Session**, then **Properties**. Select the **API** tab and put a check in the **Enable ETU** box.

If you have a DOS-based emulation product, then download IBMTRAN from ftp://ftp.nlynx.com/pub/Midrange/EmeraldSeries/ETU/IBMTRAN/ IBMT2201.EXE

OCL PROBLEMS:

PROBLEM: ETU does not work when submitted from an OCL, however it does work when entered at the command line.

CAUSE: Interactive commands can be put in a OCL script as long as they are run from the microcomputer that you are sending the file to or receiving it from.

You cannot run the interactive commands from a OCL script that is called from a batch file. This is because the underlying host commands that ETU utilizes for the interactive commands are using COPYSELF. In other words a batch file does not identify the PC. ETU sends an inquiry to (what it assumes to be) a user's microcomputer asking if it is ETU capable. Because the host sent the command, it gets the question.

SOLUTION: You must run these commands from the microcomputer that you are sending the file to or receiving it from.

PROBLEM: "**SYS1301 Invalid Procedure Name**" when running a System 36 OCL. Commands sent in the script work when sent from a command line.

CAUSE: The IBM System 36 book says: "If your are running a procedure, either you incorrectly entered a procedure name or parameter, or there is an error in the procedure. "

1) Using a batch file to send the commands.
2) There is an error in your command line.
3) CRTPF is disabled for the user name that you have logged on with.
4) The library where the file exists is not in the library list.
5) The file that you are trying to transfer is already open on the PC side.
6) User does not have ETU support on his PC emulation.

SOLUTION:
Don't use a batch file to send commands.
Don't put errors in your command line.
Enable CRTPF for your user name.
SLIB the library where the file exists.
Close the file on the PC.
Enable ETU support on your PC.

PROBLEM: "**SYS1301 Invalid Procedure Name**" when running a System 36 OCL. Commands sent in the script work when sent from a command line. OCL is started from a batch file.

CAUSE: When you execute the OCL procedure, ETU looks for a PC with ETU support, but instead it sees the System36 as the one that started the procedure and denies further action, causing your OCL to error out.

SOLUTION: You cannot start the OCL script from a batch file; it *must* be started interactively.

ERROR MESSAGES:

Some of these are not ETU messages. These are System Messages. The exact meaning could probably be traced using **F1** on the error message.

PROBLEM: Receiving the error: "**1212 I/O Error - System Output Error on Reformatted Data File**".

CAUSE: The system that you are transferring from does not have a large enough capacity for the conversion files.

SOLUTION: Change to a system with more hard drive space.
It can convert file with 236,000 records - record length 640 using *BASICS.

PROBLEM: I am getting an error message "**USER-2000 P.C. DIRECTORY IS FULL OR INVALID FILE NAME**".

CAUSE: You are trying to transfer to a directory that does not exist or the filename is invalid because it is opened by a high-level utility, perhaps Word or a virus scanner.

SOLUTION: Make certain that the directory name is valid.

If the qualified file name is incorrect, for example, the back slash leaning the wrong way, it could cause this.

Might be an antivirus program, or another program that is interfering.

It could be that another high-level utility has the file of that name open. Typical you can have that filename open in Notepad, and still transfer a file with that name, but not Word, for example.

You might just need to add an additional comma before the PC workstation file. This is a known error in the old manual.

If you are using MENU PCTRAN, try using MENU ETU36. It is newer and better. PCTRAN would require using DOS format names.

The problem could be caused by anything that does not allow a file of that name and size to be written to that directory. If the directory is temporary inaccessible, or the file locked, it could cause this. Quite often when this message comes up the application is trying to write to the A: drive.

Are you doing back-to-back file transfers? If the last transfer has not completed, it will crash one that follows too closely.

How large is the file that you are attempting to transfer? Is it possible that there is not enough room for the file on the drive that you are sending it to?

In Win95, it's using FAT32 and thus appeared to the true DOS program that the directory is full.

Make certain that you are using ES32 emulation.

PROBLEM: I am doing FROM3X on a file. Getting an error message:
"**SYS1185 Error: User One Member INVOICE file INVOICE in library QS36 not found**".

CAUSE: The Member name is still set to *FILE. This makes ETU look for a member with the name INVOICE in this case. .

SOLUTION: Change the Member name to *ALL.

PROBLEM: Running ETU 4.108 on an AS/400 in SSP emulation mode, and using the old PCTRAN commands. After MENU ETU36 brings up the menu, when you enter option 13, you get the following:

TRNSMT02
DELETE procedure is running
BLDFILE procedure is running
TRANSMIT FILE BEING CREATED

*** ERROR *** see statement:
// LOAD PCRCB2
SYS1235 Options (3)
Program PCRCB2 in LOAD statement was not found.

CAUSE: The **ETU library** is not correctly loaded.

SOLUTION:
1) Make certain that ETU is in the library it was installed in. You cannot install it and move it. There are CL's that compile with the install library name. If it was moved, you will either have to move it back or reinstall.

2) Make certain the ETU library list entries are loaded. **ADDLIBLE ETU400** (or whatever you named it).

3) You may need to remove library list entries that are incorrect. **RMVLIBLE** _____.

4) Use the newer commands of **MENU ETU36**. Use the menu entry method first. When that works, try using the command line. When your command line works, you can inert it into an OCL.

5) If it's the S/36 ENVIRONMENT, the files will usually be found in library**QS36F** (unless someone altered the default **S/36EE file library** - check with command **DSPS36** and look at **S/36 Environment Variables**).

PROBLEM: Whenever I try to transfer this file that I have transferred before, I get a message "**Invalid PC file name -- or -- No space for a new director entry.**"

CAUSE: The name that you are giving the file at the host is in use and opened by an application. There is no space left on the PC. Fat chance.
You tried to use a long file name. ETU uses the 8.3 format.

SOLUTION: Close the file that is open. Choose a name that has 8 characters with a 3-character extension. Folders must have less than 8 characters as well.

MAC PROBLEMS:

PROBLEM: I can't transfer a file to the host from my Mac. I get an error message: "**DOS error code 0C general failure probable error media error**". Transfer is run from an OCL script that asks for a filename.

CAUSE: ETU asks for a "Qualified file name", such as 'C:\PCPRINT.LST'. The filename needs to include the full path including the drive specification.
They are only typing in the filename.
Could be caused by using an incorrect path name in a script.
Could be caused by not reading the A: drive.

SOLUTION: Put an * in the qualified filename field and it will prompt the end user to look for the file. The name of the hard drive must be included in the path, and it is all case-sensitive.

PROBLEM: I want to transfer a Mac file to the host and then retrieve it. It does this, but the signature is lost.

CAUSE: ETU does not understand Mac's.

SOLUTION: Do the transfer using the **File Type** of ***SAVE** and set the **Type of Translation** to ***NONE**.

USR MESSAGES:

If a USR message that you are experiencing is not in this document, please try this procedure, on the AS/400, to get more information:

1) Go to a command line and type: **WRKMSGF ETUMSGF** and press the **<Enter>** key.

2) Take a **5** on the ETU message file.

3) Type the *error message* (USR####) in the '**Position to . . .**' field and press **<Enter>**.

Also look in the **WRKMSGF PCTM1** library.
NOTE: If the error message does not show up here, it means that ETU did not transfer the file. Almost all miscellaneous USR#### messages are usually caused by running numerous calls in a single procedure. If one call does not finish before another starts, you must add a delay between file transfer calls.

NOTE: For **CPF** messages that are not in this document follow the above procedure using the **QCPFMSG** file. For more information, check:
http://www.nlynx.com/html/tb-etu.htm
http://www.ringdale.com see the Knowledge Base.

Appendix E

ETU/36 Automation

This documentation is meant as an explanation of the considerations that one must make when one is trying to automate file transfers using ETU. It is usually easier to handle the function of automating the process on the host itself, vs. using PC macros.

Interactive vs. Batch:

Interactive means that the command strings are or appear to be typed in by a user. Batch means that the commands are sent by a utility that does not identify itself as a terminal. All ETU commands will work in interactive mode.

Interactive-only ETU Commands:

You must run these commands from the microcomputer that you are sending the file to or receiving it from.

ALOCATE, DELETEE, FILEFRPC, LIBRFRPC, LIBRTOPC, RENAME, TESTFILE, FILETOPC, PRNTTOPC, PRTFRPC, and COMPILE

Interactive commands can be put in a OCL script (called OCL on the System 36) as long as they are run from the microcomputer that you are sending the file to or receiving it from.

NOTE: You cannot run the interactive commands from a OCL script that is called from a batch file. This is because the underlying AS400 commands that ETU utilizes for the interactive commands are using COPYSELF. In other words a batch file does not identify the PC. ETU sends an inquiry to (what it assumes to be) a user's microcomputer asking if it is ETU capable. Because the AS400 sent the command, it gets the question.

Commands that can be run in batch mode:

The ETU commands that translate data or work with translation tables can be run in batch mode as well as interactively. These commands are XLTPCFIL, and XLT36FIL. You can run these commands from a microcomputer emulating a 5250 terminal or from any host terminal attached to the host.

AUTOMATION MECHANISMS:

There are OCL scripts on the System 36.

OCL scripts are created using SEU (compile). To the System 36, OCL scripts look like interactive commands entered by the terminal that submits the OCL script. They will work if the command line is formatted correctly, but if they contain 'interactive commands'; they will not work, if they are run from a batch. It is recommended that the line be formatted and tested on a command line before attempting to put it in a script.

Batch commands are submitted to a job queue. The only batch commands that will work are the translate commands.

Macros run from the ES32 emulation software will work, but the scripting mechanism does not interact, so it will not know when the file transfer is complete. So if ETU is used in a macro, it needs to be the last command of the macro.

TIPS:

Some functions must be split into two separate OCL lines. Perform the translation first, then do a transfer using reformat or type of translate = *NONE.

If you have a problem, try running the string from the menu first, the command line seconds, and then put the command line in your script.

Appendix F

CREATING F & I SPECIFICATIONS

About This Appendix

On the System 36, files are defined with Format and Input specifications (F&I). This appendix explains how to define files using F&I specs. F&I specs define field-oriented files on the System 36, which are files in either *TAB, *BASICS, or *DIF format. Before you can transfer a field-oriented file, the file must be defined with F&I specs. F&I specs are not needed when transferring straight text files (*TEXT format).

F&I specs are created using the Source Entry Utility (SEU) on the System 36. In order to use SEU you must have the proper authority. Your system administrator can verify for you whether you will be able to use SEU.

To create F&I specs for a file, you must know how each field in the file is defined. For example, you should know the field name, field length, and whether the field is numeric or alphabetic.

This appendix shows you how to create F&I specs with SEU by providing an example for you to follow.

Creating F & I Specifications

In order to transfer tabular data, like that found in spreadsheets or databases, you must create a template that tells the SYSTEM/36 where each field begins and ends, and what type of data will be in each field. This is done with Format and Input (F&I) specifications.

In this example you will see how to create F&I specs that allow a PC file to be transferred and properly formatted on the SYSTEM/36. The PC file is called SAMPLE and was created using a PC spreadsheet application. The PC spreadsheet file is represented in Figure D-1 shown below. This spreadsheet contains four fields. Table D-1, at the bottom of this page, describes how each field is defined.

Figure D-1 SAMPLE file created with PC spreadsheet

	A	B	C	D	E
1	Bart Simpson	President	45301	San Jose	
2	Homer Simpson	Vice President	34500	Austin	
3	Fred Flintstone	General Manager	57500	Denver	
4	Cliff Anderson	Technical Support Manager	42300	Springfield	
5	Tom Stiles	Customer Relations Manager	32400	Los Angeles	
6	Monica Martinez	Office Manager	24300	Buda	
7	Scott Haram	Director of Intelligence	00700	Chicago	
8	Tom Williams	Sales Manager	43200	Dallas	
9	Mr. Spock	Foreign Relations	23400	Vulcan	
10	Dilbert Dill	Engineering Manager	42300	San Jose	
11	Wally Findley	Marketing Manager	76700	Houston	
12	Herman Munster	Software Engineer	66600	Waco	
13	Cosmo Kramer	Personal Director	32100	New York	
14	Harry Potter	Vice President	43200	London	
15	Gandalf Wizard	Information Systems Manager			
16					
17					
18					
19					
20					
21					

Table D-1 Field definitions for file SAMPLE

Field Name	Field Length	Field Type	Decimal Positions
NAME	17	Alphabetic	-
TITLE	36	Alphabetic	-
SALARY	7	Numeric	2
CITY	15	Alphabetic	-

To create an F&I specification for a file, use the Source Entry Utility (SEU) on the SYSTEM/36. The steps below and on the following pages describe how to create F & I specs for the file shown on the previous page.

1. Sign on to the SYSTEM/36.

2. Enter **SEU** on the command line of the SYSTEM/36, and then press **F4**. The Start Source Entry Utility (SEU) screen, shown in Figure D-2, appears.

If you receive an error after this command, you may not have the authority to perform this task. See your system administrator.

Figure D-2 The Start Source Entry Utility screen

```
Start Source Entry Utility (SEU)

 Type choices, press Enter.

 Source file  . . . . . .   QRPGSRC          Name, *PRV
   Library  . . . . . .       QGPL        Name, *LIBL, *CURLIB, *PRV
 Source member  . . . . .   SAMPLE         Name, *PRV, *SELECT
 Source type  . . . . . .   RPG            Name, *SAME, BAS, BASP...
 Option . . . . . . . . .   *BLANK         *BLANK, ' ', 2, 5, 6
 Text 'description' . . .   *BLANK

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

3. Define the file containing the F & I specifications by entering the following information, and then press Enter.

Enter **QRPGSRC** for the source file name. QRPGSRC is a system file that contains source for F&I specs.

Enter **QGPL** for the library name. QGPL is a system library.

Enter **SAMPLE** for the source member name.

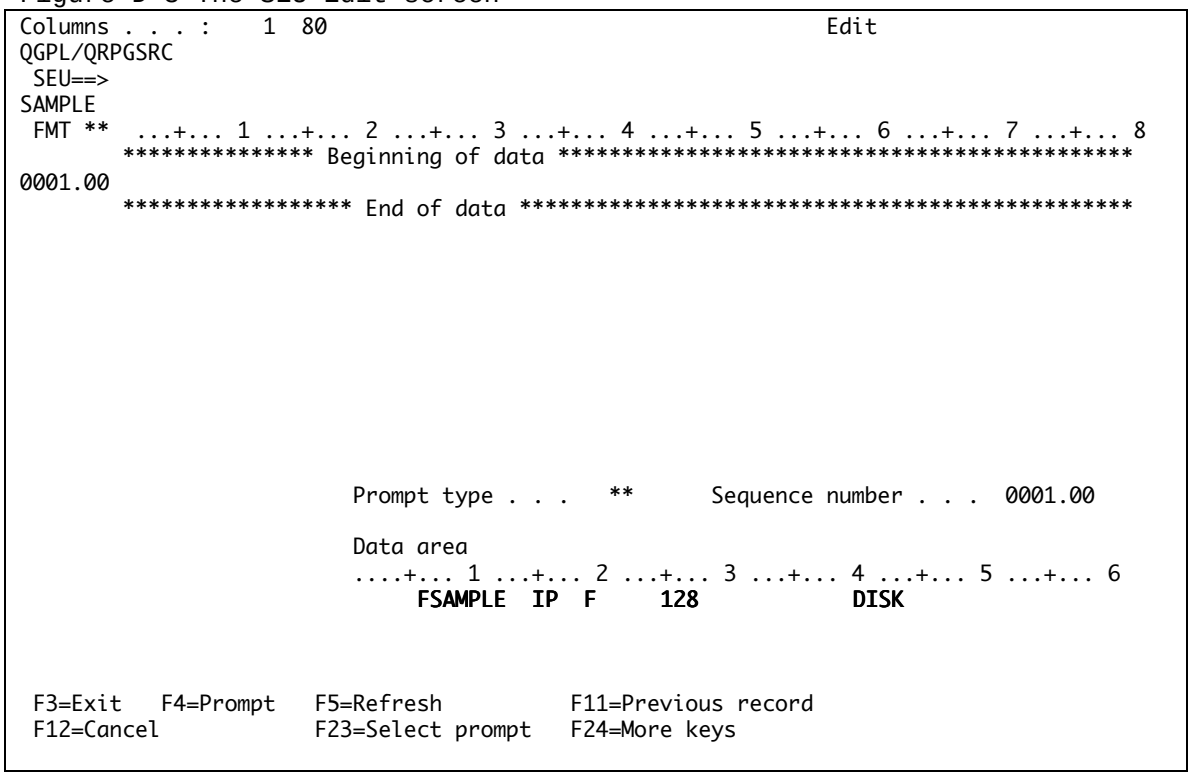
Enter **RPG** for the source type. The source for F&I specs is always RPG.

Text 'description' is an optional field you can use to identify the F&I source member.

4. When the SEU Edit screen appears (see Figure D-3), press **F4** to display a data area prompt in which to enter the file specification record. The file specification record includes the form type, the name of the file being defined, the file type, the record length, and the type of media on which the file will reside.

To create the file specification record, enter the file description specification in the data area prompt at the bottom of the screen exactly as shown in Figure D-3, and then press ENTER.

Figure D-3 The SEU Edit screen



Column 6 is the Form Type and must contain the letter **F**.

Columns 7 through 14 may contain a valid file name for documentation purposes. The file name can be the same as the PC file name, although it's not necessary that it is. This example uses file name **SAMPLE**, which in this case is the same as the PC file name.

Column 15 is the File Type and must contain the letter **I**, which defines the file as an input file.

Column 16 is the File Designation and must contain the letter **P**.

Column 19 is the File Format and must contain the letter **F**.

Columns 24 through 27 allow you to specify the record length of the data file. This example uses **128**, which is the ETU default.

Columns 40 through 46 are used to indicate where the F&I source member will be found. **DISK** means the file will reside on the hard disk of your

host system.

If you enter something incorrectly a message: "The Type entry is not H, F, E, L, I, C, O, or U (position 6)." will appear, telling you the position with an illegal value.

After you press Enter, the format specification you've just created moves to the top of the screen.

5. Display the next statement number by typing **I1** over the first two positions of line statement 0001.00 and pressing Enter.

When the new statement line is displayed, press **F4** to display a prompt in which to enter the first input specification.

Figure D-4 The SEU Edit screen for line 0002.00

```
Columns . . . :    1  80                                    Edit
QGPL/QRPGSRC
 SEU==>
SAMPLE
 FMT F  .....FFilenameIPEAF....RlenLK1AIOvKlocEDevice+......KExit++Entry+A....U1........
        *************** Beginning of data *********************************************
0001.00      FSAMPLE  IP  F     128            DISK
0002.00
        ****************** End of data ***************************************************









                        Prompt type . . .   **      Sequence number . . .  0002.00

                        Data area
                        ....+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6
                             ISAMPLE  NS



 F3=Exit   F4=Prompt   F5=Refresh          F11=Previous record
 F12=Cancel            F23=Select prompt   F24=More keys
```

6. The field statement number 0002.00 appears. Enter the input specification in the data area prompt at the bottom of the screen exactly as shown in Figure D-4, and then press ENTER.

Column 6 is the Form Type and must contain the letter **I**.

Columns 7 through 14 may contain a valid file name for documentation purposes. This example uses filename **SAMPLE**.

Type the letters **NS** for No Sequence in columns 15 and 16.

After you press Enter, the input specification you've just created moves to the top of the screen.

7. Display the next statement number by typing **I1** over the first two positions of the line containing statement 0002.00 and pressing Enter.

When the new statement line is displayed, press **F4** to display a prompt in which to enter the input specification.

8. The field statement number 0003.00 appears. Here you will define the first field in the SAMPLE file. Enter the data exactly as shown in the data area prompt at the bottom of the screen, as shown in Figure D-5, and then press ENTER.

Figure D-5 The SEU Edit screen for line 0003.00

```
Columns . . . :    1  80                                    Edit
QGPL/QRPGSRC
 SEU==>
SAMPLE
 FMT F  .....FFilenameIPEAF....RlenLK1AIOvKlocEDevice+......KExit++Entry+A....U1........
        *************** Beginning of data ********************************************
0001.00      FSAMPLE  IP  F     128            DISK
0002.00     ISAMPLE   NS
0003.00
        ***************** End of data **********************************************









                        Prompt type . . .   **      Sequence number . . .  0002.00

                        Data area
                        ....+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6
                             I                                      1  17 NAME



 F3=Exit   F4=Prompt   F5=Refresh          F11=Previous record
 F12=Cancel            F23=Select prompt   F24=More keys
```

Column 6 is the Form Type and must contain the letter **I**, which designates this record as an input specification.

Use columns 44 through 47 to enter the beginning position of the Name field. This field starts at position **1**.

Use columns 48 through 51 to enter the ending position of the Name field. This field ends in position **17**.

Use columns 53 through 58 to title the field for reference. This example calls the field **NAME**.

After you press Enter, the input specification you've just created moves to the top of the screen.

9. Display the next statement number by typing **I1** over the first two positions of the line containing statement 0003.00 and pressing Enter.

When the new statement line is displayed, press **F4** to display a prompt in which to enter the input specification.

10. The field statement number 0004.00 appears. Here you will define the second field in the SAMPLE file. Enter the data exactly as shown in the data area prompt at the bottom of screen, as shown in Figure D-6, and then press ENTER

Figure D-6 SEU Edit screen for line 0004.00

```
Columns . . . :    1  80                                      Edit
QGPL/QRPGSRC
 SEU==>
SAMPLE
 FMT F  .....FFilenameIPEAF....RlenLK1AIOvKlocEDevice+......KExit++Entry+A....U1........
       *************** Beginning of data ********************************************
0001.00      FSAMPLE  IP  F     128            DISK
0002.00      ISAMPLE  NS
0003.00      I                                      1  17 NAME
0004.00
       ****************** End of data ***************************************************









                        Prompt type . . .   **      Sequence number . . .  0004.00

                        Data area
                        ....+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6
                             I                                      18  52 TITLE



 F3=Exit   F4=Prompt   F5=Refresh          F11=Previous record
 F12=Cancel            F23=Select prompt   F24=More keys
The Type entry is not H, F, E, L, I, C, O, or U (position 6).
```

Column 6 is the Form Type and must contain the letter **I**, which designates this record as an input specification.

Use columns 44 through 47 to enter the beginning position of the Job Title field. This field starts at position **18**.

Use columns 48 through 51 to enter the ending position of the Job Title field. This field ends in position **52**.

Use columns 53 through 58 to title the field for reference. This example calls the field **TITLE**.

After you press Enter, the input specification you've just created moves to the top of the screen.

11. Display the next statement number by typing I1 over the first two positions of the line containing statement 0004.00 and pressing Enter.

When the new statement line is displayed, press F4 to display a prompt in which to enter the input specification.

12. The field statement number 0005.00 screen appears. Here you will define the third field in the SAMPLE file. Enter the data exactly as shown in the data area prompt at the bottom of screen, as shown in Figure D-7, and then press ENTER.

Figure D-7 SEU Edit screen for line 0005.00

```
Columns . . . :    1  80                                    Edit
QGPL/QRPGSRC
 SEU==>
SAMPLE
 FMT F  .....FFilenameIPEAF....RlenLK1AIOvKlocEDevice+......KExit++Entry+A....U1........
       ********** Beginning of data ***********************************************
0001.00      FSAMPLE  IP  F     128            DISK
0002.00      ISAMPLE  NS
0003.00      I                                      1  17 NAME
0004.00      I                                     18  52 TITLE
0005.00
       ****************** End of data ***************************************************




                        Prompt type . . .   **      Sequence number . . .  0005.00

                        Data area
                        ....+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6.
                             I                                       53  60SALARY



 F3=Exit   F4=Prompt   F5=Refresh          F11=Previous record
 F12=Cancel            F23=Select prompt   F24=More keys
```

Column 6 is the Form Type and must contain the letter **I**, which designates this record as an input specification.

Use columns 44 through 47 to enter the beginning position of the Salary field. This field starts at position **53**.

Use columns 48 through 51 to enter the ending position of the Salary field. This field ends in position **60**.

Column 52 specifies the decimal places of the numeric field. Use column 52 if the field contains numeric data and you would like to specify the number of decimal positions. For our example, SALARY, you will use **2** decimal positions.

Use columns 53 through 58 to title the field for reference. This example calls the field **SALARY**.

After you press Enter, the input specification you've just created moves to the top of the screen.

13. Display the next statement number by entering **I1** in the first two positions of the line containing statement 5 and pressing Enter.

When the new statement line is displayed, press **F4** to display a prompt in which to enter the input specification.

14. The field statement number 0006.00 screen appears. Here you will define the third field in the SAMPLE file. Enter the data exactly as shown in the data area prompt at the bottom of screen, as shown in Figure D-8, and then press ENTER.

Figure D-8 SEU Edit screen for line 0006.00

```
Columns . . . :    1  80                                    Edit
QGPL/QRPGSRC
 SEU==>
SAMPLE
 FMT F  .....FFilenameIPEAF....RlenLK1AIOvKlocEDevice+......KExit++Entry+A....U1........
        *************** Beginning of data ***********************************************
0001.00      FSAMPLE  IP  F     128            DISK
0002.00      ISAMPLE  NS
0003.00      I                                       1  17 NAME
0004.00      I                                      18  52 TITLE
0005.00      I                                      53  602SALARY
0006.00
        ***************** End of data ***************************************************




                         Prompt type . . .   **      Sequence number . . .  0006.00

                         Data area
                         ....+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6
                              I                                      61  76 CITY



 F3=Exit   F4=Prompt   F5=Refresh          F11=Previous record
 F12=Cancel            F23=Select prompt   F24=More keys
The Type entry is not H, F, E, L, I, C, O, or U (position 6).
```

Column 6 is the Form Type and must contain the letter **I**, which designates this record as an input specification.

Use columns 44 through 47 to enter the beginning position of the City field. This field starts at position **61**.

Use columns 48 through 51 to enter the ending position of the City field. This field ends in position **76**.

Use columns 53 through 58 to title the field for reference. This example calls the field **CITY**.

If you need to make any corrections to any field or input specifications, move the cursor to the appropriate line and press **F4**.

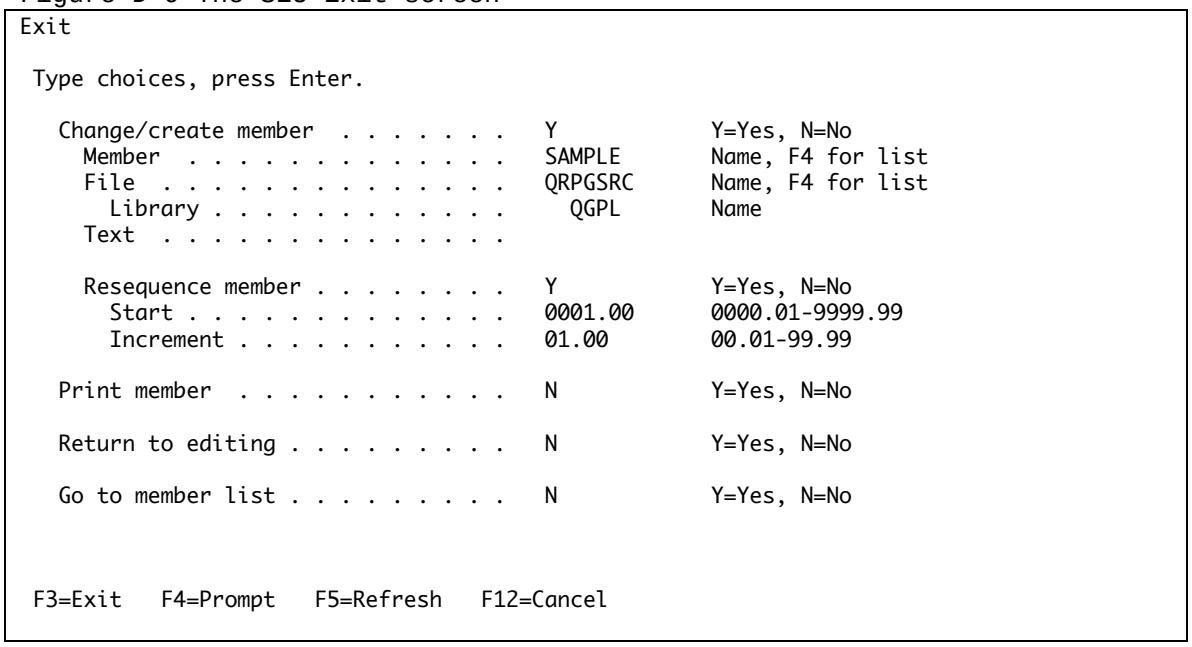
You will notice that the prompt screen now shows the field position names for the line type you have selected with the sequence number highlighted. Press the Tab key to move to the field to be changed, make any appropriate changes, and press ENTER.

If you need to insert a line, enter I1 in the first two positions of the line number that will precede the new line. For example, enter I1 over line number 0003.00 to insert a new line following line 0003.00.

15. Press **F3** to complete the F & I specification. The SEU Exit screen appears as shown in Figure D-9. Verify that the Change/create member parameter is Y for yes. Also verify that the member name and the library name on this screen are correct, and then press ENTER. You will be returned to the SYSTEM/36 screen you were at prior to issuing the SEU command.

The F&I specification called SAMPLE for the demo file called ETUFILE is now defined. You can now use this F&I definition to transfer the file using ETU.

Figure D-9 The SEU Exit screen



For more information about F&I specifications, please refer to the IBM SYSTEM/36 Languages: Systems Application Architecture AD/Cycle RPG/400 Reference Manual SC09-1349-00

Appendix G

GLOSSARY

Allocate	To create a file on the microcomputer.
API	Application Program Interface. The interface (calling conventions) by which an application program accesses operating system and other services.
ASCII character set	American Standard Code for Information Interchange. A standard set of characters used by microcomputers to represent data. See also EBCDIC character set.
BASIC sequential format	One of four micro file formats supported by ETU. BASIC sequential files are usually created with database applications. Commas separate the fields within a BASIC sequential file. The file extension on the PC is .CSV (comma separated values).
batch processing	One of two ways of running jobs on the AS/400 (along with interactive processing). A job run in batch is submitted to a job queue where it will be processed when it reaches the top of the queue.
comma-separated values	This is a file format used on PC's where commas separate the fields. This is also referred to as a BASIC sequential file. The file extension on the PC is .CSV (comma separated values).
data transfer	The process of moving data between the micro and the host with ETU.
data translation	An ETU process in which data is changed from ASCII to EBCDIC or from EBCDIC to ASCII, so that host data can be used on the micro, and micro data can be used on the host.
DIF format	One of four micro file formats supported by ETU. DIF files are usually created with spreadsheet applications such as LOTUS 1-2-3.
EBCDIC character set	Extended Binary Coded Decimal Interchange Code. A set of characters used by IBM midrange computers and display stations to represent data. See also ASCII character set.

ETU	Emulator Transfer Utility. AS/400 utility that transfers files to or from 5250 emulators. The best around.
F&I	Format and Input specifications. A means of creating internal file definitions on the AS/400. ETU supports F&I specs.
host system	The System/36 or Advanced/36 computer and its peripherals.
IBMTRAN	An interface program for PCs that is included with ETU. It allows non-ES32 emulation packages to be used with ETU.
interactive processing	One of two ways of running jobs on the AS/400 (along with batch processing). A job that is run interactively begins being processed the instant it is executed from the workstation. The workstation is tied up for other purposes until the job is complete.
job queue	A waiting area on the AS/400 where batch jobs are sent to wait for processing. Jobs in the job queue are processed in the order that they were received in the queue.
load member	A library member that contains information in machine language, a form that the system can user directly. Contrast with source member. Also called object files.
logical file	A description of how data is to be presented to or received from a host program. This type of file contains no data, but it provides an ordering and format for one or more physical files. Contrast with physical file.
object file	A passive entity that contains or receives information but cannot change the information it contains. For DB2 objects include rows, tables, databases, stored procedures, triggers, defaults, and views.
object code	Output from a compiler or assembler that is also executable machine code or is suitable for processing to produce executable machine code. Contrast with source code.
OCL	Object Constraint Language. A modeling language used to set up strings of valid SSP commands.
output queue	A waiting area on the AS/400 where host spool files are sent until they are released to a printer. Print jobs on the output queue are processed in the order that they were received in the queue.

physical file A description of how data is actually stored on the host. A physical file contains one record format and one or more members. Contrast with logical file.

source code Source for source members. source code The input to a compiler or assembler, written in a source language. Contrast with object code.

source language A language from which statements are translated.

SSP (**S**ystem **S**upport **P**rogram) A multi-user, multitasking operating system from IBM that is the primary control program for the System/36.

System/36 mode A set of commands and procedures within the AS/400 operating system that simulates the operating system of the System/36. It allows users of the System/36 to work on the AS/400 without first learning the operating system of the AS/400.

Tab format *TAB. One of four micro file formats supported by ETU. Tab files are usually created with database or applications. The fields in a TAB formatted file are separated by tab characters. The file extension on the PC should be .TXT.

Text format *TEXT. One of four micro file formats supported by ETU. Text files are usually created using word processors and must be converted to ASCII, the lowest common denominator of microcomputer data, before being transferred to the host.

Appendix H

INDEX

*BASICS, 1, 14, 17, 20, 24, 33, 38, 2, 14
*TAB, 14, 17, 20, 24, 33, 38, 3, 1, 2, 3, 4, 5, 7, 3
ALLOCATE, 2, 1, 7, 8
BASIC Sequential, 1, 14, 17, 20, 24, 33, 38, 2, 14
Compatibility, 3, 4, 1
COMPILE, 1, 5, 7, 9, 12, 14, 20, 25, 33, 4, 1
EDITABLE, 1, 5, 7, 11, 14, 15, 17, 20, 21, 25, 33, 38, 10, 13
emulation, 4, 5
emulator, 4, 5
F & I, 1, 2, 3, 7, 8, 9, 1, 2, 3, 11
Field & Index, 1, 2, 3, 7, 8, 9, 1, 2, 3, 11
FILEFRPC, 2, 1, 6, 7, 13, 14, 8, 1
FILETOPC, 2, 3, 1, 4, 5, 7, 14, 16, 17, 20, 24, 33, 38, 14, 3, 6, 1
IBMTRAN, 3, 4, 1, 4, 5, 2
interactive, 2, 1, 9, 1, 2, 1, 2
LIBRFRPC, 2, 1, 5, 7, 19, 1
LIBRTOPC, 2, 3, 1, 7, 23, 1
Mac, 4, 1, 2, 1, 2, 3, 6, 9, 10, 12, 14, 15, 17, 20, 21, 24, 25, 26, 28, 30, 33, 38, 40, 6
Mac, 4, 1, 2, 1, 2, 3, 6, 9, 10, 12, 14, 15, 17, 20, 21, 24, 25, 26, 28, 30, 33, 38, 40, 6
PRNTFRPC, 2, 1, 7, 26
PRNTTOPC, 2, 3, 1, 28, 14, 1
tab, 14, 17, 20, 24, 33, 38, 3, 1, 2, 3, 4, 5, 7, 3
TCP/Axcess, 2, 3, 2, 1, 4, 5
TESTFILE, 2, 3, 1, 7, 31, 14, 1
workstation file name, 2, 6, 8, 10, 13, 16, 18, 19, 23, 25, 26, 28, 29, 30, 31
XLT36FIL, 1, 5, 7, 32, 1
XLT36PRT, 1, 7, 35
XLTPCFIL, 1, 5, 7, 37, 1
XLTPCPRT, 1, 7, 40

Technical Support

North America
(8:00 a.m. to 5:00 p.m. CST, Monday to Friday)
Fax: 512-930-1699
Phone: 888-288-9080
512-288-9080

Europe
(0900 to 1700 GMT, Monday to Friday)
Fax: +44 161 928 7015
Phone: +44 161 928 7014

Please send manual comments to Support@Ringdale.com