



# **ETU400**

## **User Guide**

*AS/400 File Transfer Utility*

# *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. Nlynx 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:

- \* "Nlynx" means Nlynx Technologies, Inc. acting by and through one or more of its subsidiaries in its Network Systems group.
- \* "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

=====

Nlynx 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, unless the Installation Manual specifies that this Software may be used on a greater number of computers.
- \* Make one backup copy of the Software, which automatically becomes the property of Nlynx 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 N Lynx, transfer all (but not a lesser portion) of the Software and related manuals to another person or entity, who 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 make any 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 N Lynx 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 N Lynx and others. N Lynx 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 N Lynx's rights. N Lynx 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 N Lynx.

## LIMITED WARRANTY

=====

NLynx warrants that, upon delivery by NLynx, 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, NLynx 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 NLynx 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) NLYNX 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) NLYNX WILL NOT BE LIABLE FOR SPECIAL, INCIDENTAL, CONSEQUENTIAL, INDIRECT OR OTHER SIMILAR DAMAGES EVEN IF NLYNX OR ITS AGENT HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS MEANS NLYNX IS NOT RESPONSIBLE OR LIABLE FOR DAMAGES OR COSTS INCURRED AS A RESULT OF LOSS OF TIME, LOSS OF DATA, LOSS OF PROFITS OR REVENUE, OR LOSS OF USE OF THE SOFTWARE. IN ADDITION, NLYNX IS NOT RESPONSIBLE OR LIABLE FOR DAMAGES OR COSTS INCURRED IN CONNECTION WITH OBTAINING SUBSTITUTE SOFTWARE, CLAIMS BY OTHERS, INCONVENIENCE OR SIMILAR COSTS;

(C) NLYNX 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 NLYNX, SUCH AS DEALER ADVERTISING OR PRESENTATIONS, DO NOT CONSTITUTE WARRANTIES OR CONDITIONS BY NLYNX, DO NOT BIND NLYNX AND SHOULD NOT BE RELIED UPON.

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

Though NLynx is not responsible for maintaining or helping you use the Software, NLynx 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 NLynx and You concerning the contents of this Package. NLynx is not bound by any provision of any purchase order, receipt, acceptance, confirmation, correspondence, or otherwise, unless NLynx specifically agrees to the provision in writing. Any written or oral information or advice given by NLynx 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 NLynx.

## Legal Notices

This document is for use with the N Lynx Technologies ETU/400 product.

No part of this document may be copied in any form or by any means without prior written consent of N Lynx Technologies. All software, firmware and downloadable operating code described herein or used by the products described herein are copyrighted by and shall remain the property of N Lynx Technologies.

All drawings, schematics and artwork used in the manufacture of products described herein are copyrighted. Reproduction of said drawings, schematics, and artwork, or manufacture of said products without written consent of N Lynx Technologies is absolutely prohibited.

### ***Warning:***

This product is subject to export controls by the United States Department of Commerce and may not be exported from the U.S. without prior written permission of the U.S. Department of Commerce Bureau of Export Administration

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.

January 2003 Edition

This edition applies to version 5.6 of ETU 400, 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:  
N Lynx Technologies  
8313 Highway 71 West  
Austin, Texas 78735

Toll-free 1-888-N Lynx67 (Technical Support Hotline)  
Technical Support is provided for up to one year from the date of purchase. Serial number is required. Technical Support by telephone is from 8:30 - 5:30 CST. We will provide answers and support by email for product that is not under warranty if you use the Technical Support Contact Form at <http://www.nlynx.com/html/contacttechnicalsupport.htm>

Technical Bulletins for ETU are located at <http://www.nlynx.com/html/tb-etu.htm>

*Printed in USA – Document 401–9140–17 Rev. B – January 2003*

**© 2000 N Lynx Technologies, Inc. All rights reserved.**

## PREFACE

This manual is intended for users of the ETU 400 Emulator Transfer Utility. It assumes the reader has a basic working knowledge of the IBM AS/400, the IBM Personal Computer or Personal System/2, and the operation of software for emulation of the IBM 5251 Model 11 or Model 12 workstations on the IBM Personal Computer. We recommend the user read, and have available for reference, the following publications:

- AS/400 Languages: RPG/400 Reference Manual, SC0 -1089
- AS/400 Programming: Control Language Reference Manual, SC21-9755, SC21-9776, SC21-9777, SC21-9778, SC21-9779
- AS/400 Programming: Data Description Specifications Reference, SC-21-9620
- AS/400 Security: Concepts and Planning, SC21-8083
- AS/400 SEU User's Guide/Reference, SC09-1172

### **Trademark Recognition.**

Throughout this manual the terms IBM, IBM PC, IBM PC/XT, IBM PC-AT, IBM PS/2, AS/400, OS/400, and IBM Enhanced 5250 Emulator are used in explanations and instructions. These terms are registered trademarks of International Business Machines Corporation.

3XTwin, 3XTwin/2, 3XPlus, 3XPlus/2, 3XMate, 3XMate/2, 3XMate-Laptop, Multi-Twin, MacTwin, and 3XTwin-Laptop are registered trademarks of Emerald Technology, Inc., an N Lynx Technologies company.

ETU is a trademark of Emerald Technology, Inc., an N Lynx Technologies company.

InterAcess is a trademark of KMW, an N Lynx Technologies company.

DIF is a trademark of Software Arts, Inc.

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

Mac is a registered trademark of Apple Corporation.

## About This Guide

Use this guide to install and run ETU 400 on the IBM AS/400 host computer.

### Who Should Use This Guide?

All users of ETU, from the beginning user of the AS/400 to the most advanced user, should refer to this guide. ETU interfaces directly with the 5251 emulation software running on your microcomputer, whether it's an IBM PC, an Apple Mac or a UNIX workstation. This guide assumes that you have a working knowledge of your microcomputer and the emulation software that resides on it.

### How This Guide is Organized

This guide is divided into four parts. Each part is briefly described below.

#### **Part One: Getting Started**

Part One is intended for all users of ETU, and should be read prior to installation. It lists the items you should have received with ETU, the host and microcomputer hardware and software requirements for running ETU, and installation instructions.

#### **Part Two: Running ETU**

Part Two is intended for all users, but especially for first-time users of ETU. It describes the basic concepts and terms used with ETU and the AS/400. In addition, it explains the most common ETU commands and provides examples of how to use them.

#### **Part Three: ETU Command Reference**

Part Three is like a reference guide for ETU. It is intended for experienced users of both ETU and the AS/400. You will find all ETU commands listed in alphabetical order, along with the functions and the parameters associated with each command.

#### **Part Four: Appendices**

The appendices include supplementary technical information, followed by an index.

Legal Notices	v
© 2000 N Lynx Technologies, Inc. All rights reserved.	v
<b>PREFACE</b>	<b>v</b>
<b>Trademark Recognition.</b>	<b>vi</b>
<b>About This Guide</b>	<b>vii</b>
Who Should Use This Guide?	vii
How This Guide is Organized	vii
<b>INTRODUCTION</b>	<b>1-1</b>
<b>Welcome to ETU</b>	<b>1-2</b>
What ETU Can Do for You	1-2
Inventory Checklist	1-3
Security Considerations	1-3
<b>Hardware and Software Requirements</b>	<b>1-4</b>
Hardware Requirements	1-4
Software Requirements	1-4
<b>Features of Version 5.02</b>	<b>1-5</b>
<b>Features of Version 5.041</b>	<b>1-6</b>
<b>Features of Version 5.5</b>	<b>1-7</b>
<b>Features of Version 5.6</b>	<b>1-8</b>
Notes to Mac Users	1-8
Compatibility with Non-ES32 Emulation Products	1-8
Native Mode and System/36 Mode	1-8
<b>Part One, Getting Started</b>	<b>1-9</b>
<b>INSTALLING ETU</b>	<b>2-1</b>
<b>About This Chapter</b>	<b>2-2</b>
<b>Installing ETU</b>	<b>2-3</b>
Installing the ETU Software	2-3
Registering Your Host System	2-6
Using ETU During the Evaluation Period	2-6
Obtaining Your Host Registration Number	2-7
Entering the Registration Code	2-7
Re-installing ETU	2-7
Removing ETU From Your Host System	2-8
Installing ETU For International Users	2-8
<b>The ETU Compatibility Diskette: For Non-ES32 Emulation Products</b>	<b>2-9</b>
Who Could Use IBMTRAN?	2-9
Copying IBMTRAN To Your Hard Drive	2-9
Starting ETU When Using IBMTRAN	2-10
Exiting IBMTRAN	2-10
Alternative Method to Start and Exit IBMTRAN	2-10



<b>USING ETU</b>	<b>3-1</b>
<b>About This Chapter</b>	<b>3-2</b>
<b>Starting ETU</b>	<b>3-3</b>
A Note About User Authorization	3-3
<b>The ETU Main Menu</b>	<b>3-4</b>
The ETU Utilities Menu	3-5
<b>The ETU Commands</b>	<b>3-6</b>
Main Menu: File Transfer Commands	3-6
Main Menu: ETU Power Commands	6
Utilities Menu: Microcomputer Commands	6
Utilities Menu: File Translation Commands	6
Utilities Menu: Translation Table Commands	6
<b>Running ETU Commands</b>	<b>3-7</b>
Using the Menu	3-7
Entering Commands on the Command Line	3-7
Using the Command Prompts	3-8
Using Mac Directory Dialog Boxes	3-8
Request Definitions	3-9
<b>Types of Processing</b>	<b>3-10</b>
Interactive ETU Commands	3-10
Batch ETU Commands	3-10
<b>About Microcomputer Files</b>	<b>3-11</b>
File Formats	3-11
PC-DOS	3-11
Apple Mac	3-11
<b>LEARNING ABOUT ETU</b>	<b>4-1</b>
<b>About This Chapter</b>	<b>4-2</b>
<b>What is ETU?</b>	<b>4-3</b>
Data Transfer	4-3
Data Translation	4-3
Other ETU Functions	4-3
<b>About Transferring Files</b>	<b>4-4</b>
Micro and Host File Names	4-4
Transferring Data to Host Files	4-5
Record Length	4-5
Host File Type	4-5
Data Description Specifications	4-6
<b>About Translating Files</b>	<b>4-7</b>
How Data is Stored	4-7
<b>Using Translation Tables</b>	<b>4-8</b>
ETU Translation Table Members	4-9
Using Translation Tables with the Mac	4-10
<b>Using Microcomputer Data Formats</b>	<b>4-12</b>

5-1	Transferring Files	3
6-1	Translating Files	3
7-1	Working with PC Functions	3
8-1	Modifying Translation Tables	3
9-1	Advanced ETU Features	3

## **TRANSFERRING FILES** **5-1**

### **About This Chapter** **5-2**

#### **Transferring Data Files** **5-3**

Transferring Data Files From Host to a Microcomputer:	5-3
FROM3XB Example	5-3
Transferring Data Files From a Microcomputer to the Host:	5-5
The TO3XB Command	5-5
TO3XB Example	5-6

#### **Transferring Print Files** **5-7**

Transferring a Host Spool File to a PC Print File:	5-7
About Host Spool Files	5-7
Transferring a PC Print File to the Host:	5-10
PRTTO3X Example	5-11

#### **Transferring Host Programs** **5-12**

Transfer Host Programs to a Microcomputer:	5-12
The PGMFROM3XB Command	5-12
PGMFROM3XB Example	5-13
Transfer Host Programs to the Host:	5-14

## **TRANSLATING FILES** **6-1**

### **About This Chapter** **6-2**

#### **Why Use the Translation Commands** **6-3**

Translation Commands Can Run in Batch Mode	6-3
Translation Commands Can Be Run From a Host Terminal	6-3
Translation Commands Reduce Modem Time	6-3

#### **Translating Data Files** **6-4**

Translating a Microcomputer File to a Host Format:	6-4
The XLATEFROM Command	6-4
XLATEFROM Example	6-5
Translating a Host File to Microcomputer Format:	6-7
XLATETO Example	6-8

#### **Translating Print Files** **6-9**

Translating PC Print Files:	6-9
The PRTXFRMPC Command	6-9
PRTXFRMPC Example	6-9
Translating Host Spool Files:	6-11
The PRTXTOPC Command	6-11
PRTXTOPC Example	6-12

## **WORKING WITH PC FUNCTIONS** **7-1**

Rename a Micro File:	7-3
The RENAMEB Command	7-3

RENAMEB Example	7-3
Figure 7-1 The RENAMEB prompt screen	7-3
Allocate a Micro File:	7-4
The ALOCATB Command	7-4
ALOCATB Example	7-4
Delete an Existing File:	7-5
The DELETEDB Command	7-5
DELETEDB Example	7-6
Test for the Existence of a Micro File:	7-7
The TESTB Command	7-7
TESTB Example	7-7
<b>MODIFYING TRANSLATION TABLES</b>	<b>8-1</b>
<b>About This Chapter</b>	<b>8-2</b>
<b>About Translation Tables</b>	<b>8-3</b>
<b>Modifying a Translation Table</b>	<b>8-4</b>
Make a Copy of the Original Table	8-5
Editing the Translation Table	8-5
Translation Table Record Format	8-5
Sample Translation Table	8-9
Compiling the New Translation Member	8-10
Naming the New Member	8-10
<b>ADVANCED ETU FEATURES</b>	<b>9-1</b>
<b>About This Chapter</b>	<b>9-2</b>
<b>Multiple-device Transfer</b>	<b>9-3</b>
Transferring From Multiple Micros	9-3
Using PWRT03XB	9-5
Transferring a Host File to Multiple Micros	9-6
Micro-to-micro Transfers	9-7
The PCFROMPC Command	9-7
The PCTOPC Command	9-9
<b>Transferring Files Between Multiple Macs</b>	<b>9-10</b>
Mac File Transfer Examples	9-10
If InterAccess is located at the root level of the hard drive	9-10
If InterAccess is one or more folders deep	9-11
<b>Record and Field Selection</b>	<b>9-12</b>
About Creating Auxiliary Storage Pools	9-12
Transferring Selected Fields	9-12
Field Selection -- Example 1	9-14
Field Selection -- Example 2	9-15
Transferring Selected Records	9-16
Record Selection -- Example 1	9-17
Record Selection -- Example 2	9-17
<b>Entry and Exit Programs</b>	<b>9-18</b>
Example	9-18
<b>Deleting Files After Transfer</b>	<b>9-19</b>

<b>Saving Request Definitions</b>	<b>9-20</b>
Re-using Request Definitions	9-20
<b>Performing Batch Transfers</b>	<b>9-22</b>
Batch Transfer – Example	9-23
<b>Time and Date Scheduling</b>	<b>9-24</b>
Time and Date Scheduling – Example	9-25
<b>ETU COMMAND REFERENCE</b>	<b>10-1</b>
<b>ALOCATB Command</b>	<b>10-2</b>
<b>DELETEB Command</b>	<b>10-3</b>
<b>EDITTABLE Command</b>	<b>10-4</b>
<b>EDTUDF Command</b>	<b>10-6</b>
EDTUDF Notes	10-7
<b>FROM3XB Command</b>	<b>10-8</b>
FROM3XB Notes	10-11
<b>PCFROMPC Command</b>	<b>10-12</b>
<b>PCTOPC Command</b>	<b>10-14</b>
<b>PGMFROM3XB Command</b>	<b>10-16</b>
<b>PGMTO3XB Command</b>	<b>10-17</b>
<b>PRTFROM3X Command</b>	<b>10-18</b>
<b>PRTTO3X Command</b>	<b>10-20</b>
<b>PRTXFRMPC Command</b>	<b>10-22</b>
<b>PRTXTOPC Command</b>	<b>10-24</b>
<b>PWRDELETEB Command</b>	<b>10-27</b>
<b>PWRFROM3XB Command</b>	<b>10-28</b>
<b>PWRRENAMEB Command</b>	<b>10-34</b>
<b>PWRTO3XB Command</b>	<b>10-35</b>
<b>RENAMEB Command</b>	<b>10-41</b>
<b>SBMBATXFER Command</b>	<b>10-42</b>
<b>SBMSCHJOB Command</b>	<b>10-43</b>
Before You Begin	10-43
To Schedule a Job	10-44
<b>STRSEU Command</b>	<b>10-45</b>

<b>TESTB Command</b>	<b>10-47</b>
<b>TO3XB Command</b>	<b>10-48</b>
<b>VWSAVRQS Command</b>	<b>10-52</b>
VWSAVRQS Notes	10-52
<b>XLATEFROM Command</b>	<b>10-54</b>
<b>XLATETO Command</b>	<b>10-57</b>
A-1    Inside ETU	10-61
B-1    ETU Compatibility	10-61
C-1    Creating DDS Specifications	10-61
D-1    Creating F & I Specifications	10-61
E-1    ETU Error Messages	10-61
F-1    EBCDIC/ASCII Printable Characters	10-61
G-1    System/36-mode Commands	10-61
H-1    International Translational Tables	10-61
GLOSSARY	10-61
INDEX	10-61
Appendix A	A-1
<b>INSIDE ETU</b>	<b>A-1</b>
<b>About This Chapter</b>	<b>A-2</b>
<b>Editing User-defined Menu Options</b>	<b>A-3</b>
EDTUDF Parameters	A-4
EDTUDF Example	A-5
<b>Configuring the Audit Trail</b>	<b>A-6</b>
The CFGAUDIT Parameters	A-7
Checking the Audit Trail	A-8
Working with ETU Security	A-9
Granting Authority to a Command	A-9
Revoking Authority From a Command	A-10
Regranting Authority to a Previously Revoked Command	A-10
<b>Securing Devices on the AS/400</b>	<b>A-11</b>
<b>Removing ETU From the AS/400</b>	<b>A-12</b>
<b>APPENDIX C</b>	<b>B-1</b>
<b>ETU COMPATIBILITY</b>	<b>B-1</b>
<b>About This Appendix</b>	<b>B-2</b>
<b>Using ETU 5.01 and above with Autokey Batch Files</b>	<b>B-3</b>
Modifying an Autokey Batch File – Example	B-3
<b>CREATING DDS SPECIFICATIONS</b>	<b>C-1</b>
<b>About This Appendix</b>	<b>C-2</b>

Creating DDS Specifications	C-3
<b>APPENDIX D</b>	<b>D-1</b>
<b>CREATING F&amp;I SPECIFICATIONS</b>	<b>D-1</b>
About This Appendix	D-2
Creating F & I Specifications	D-3
Appendix E	E-1
<b>ERROR MESSAGES: NATIVE MODE</b>	<b>E-1</b>
About This Appendix	E-2
<b>USR MESSAGES:</b>	<b>E-3</b>
<b>TRANSFER PROBLEMS:</b>	<b>E-4</b>
<b>USER MESSAGES:</b>	<b>E-5</b>
<b>CPF MESSAGES:</b>	<b>E-15</b>
<b>SYS MESSAGES:</b>	<b>E-19</b>
<b>OCL PROBLEMS</b>	<b>E-21</b>
Appendix F	F-1
<b>ASCII-EBCDIC PRINTABLE CHARACTERS</b>	<b>F-1</b>
<b>About This Appendix</b>	<b>F-2</b>
This appendix lists the ASCII and EBCDIC characters and their hexadecimal equivalents. Use this table when you create or modify an ETU translation table.	F-2
Appendix G	G-1
<b>SYSTEM/36-MODE COMMANDS</b>	<b>G-1</b>
About This Appendix	G-2
<b>Starting ETU in System/36 Mode</b>	<b>G-3</b>
The ETU36 Menu	G-3
Running ETU in System/36 Mode	G-5
Native Commands Not Supported in the System/36 Mode	G-5
System/36-mode Commands and Native Mode Equivalents	G-5
<b>ALLOCATE Procedure</b>	<b>G-6</b>
<b>COMPILE Procedures</b>	<b>G-7</b>
<b>DELETE Procedure</b>	<b>G-8</b>
<b>EDITABLE Procedure</b>	<b>G-9</b>

<b>FILEFRPC Procedure</b>	<b>G-10</b>
<b>FILETOPC Procedure</b>	<b>G-13</b>
<b>LIBRFRPC Procedure</b>	<b>G-15</b>
<b>LIBRTOPC Procedure</b>	<b>G-18</b>
<b>PRNTFRPC Procedure</b>	<b>G-21</b>
<b>PRNTTOPC Procedure</b>	<b>G-23</b>
<b>RENAME Function</b>	<b>G-25</b>
<b>TESTFILE Procedure</b>	<b>G-26</b>
<b>XLT36FIL Procedure</b>	<b>G-27</b>
<b>XLT36PRT Procedure</b>	<b>G-29</b>
<b>XLTPCFIL Procedure</b>	<b>G-31</b>
<b>XLTPCPRT Procedure</b>	<b>G-33</b>
Appendix H	H-1
 <b>INTERNATIONAL TRANSLATION TABLES</b>	 <b>H-1</b>
<b>Installing International Translation Table</b>	<b>H-2</b>
Appendix I	I-1
 <b>ETU AUTOMATION</b>	 <b>I-1</b>
<b>ETU400 Automation</b>	<b>I-1</b>
AUTOMATION MECHANISMS	I-2
SBMBATXFER	I-2
SBMSCHJOB	I-2
 <b>ETU THROUGHPUT CONSIDERATIONS</b>	 <b>J-1</b>
<b>ETU THROUGHPUT CONSIDERATIONS</b>	<b>J-2</b>
What kind of throughput can I expect?	J-2

*Part One*

# Getting Started

"Getting Started" is intended for all users of ETU. It describes how to install and how to use ETU. If you are new to ETU, you will be introduced to concepts and terms that will help you master it.

- 2-1                    Installing ETU**
- 3-1                    Using ETU**
- 4-1                    Learning About ETU**



# INTRODUCTION

Welcome to ETU

- What ETU Can Do For You
- Inventory Checklist
- Security Considerations

Hardware and Software Requirements

- Hardware Requirements
- Software Requirements

Features of Versions

- Notes to Mac Users
- Compatibility with Non-NLynx ES32 products
- Native Mode and System/36 Mode

## Welcome to ETU

ETU is a host software package that allows you to transfer files between an IBM AS/400 host computer and one or more microcomputers that are emulating an IBM 5250 display station.

### What ETU Can Do for You

The purpose of ETU is twofold: to transfer data and to translate data. Data transfer entails the movement of data between two systems.

Data translation changes the way the data is represented or stored so that the same data can be used by two different systems. More specifically, with ETU, you can:

- Transfer data between your microcomputer and the AS/400;
- Transfer data from one microcomputer to another microcomputer of the same type;
- Transfer user-specified fields or records between the host and the microcomputer;
- Transfer a single host file to as many as ten microcomputers with one command;
- Translate files to and from the following microcomputer formats: ASCII TEXT, BASIC Sequential, DIF, and TAB;
- Create, rename, test for existence of, and delete files on the micro while running 5250 emulation from the microcomputer;
- Use the host for off-line storage of microcomputer files.

Table 1-1 \* ETU functions and their required security status

ETU Function	Security Profile Required
Install ETU	QSECOFR
Use all ETU commands (other than the power commands)	*PUBLIC
Use ETU Power commands	QSECOFR, or a user profile with *ALLOBJ authority, or any user who has been granted explicit authority to a power command
Configure Audit Trail	QSECOFR or a user profile with *ALLOBJ authority

## Inventory Checklist

In addition to this manual, the following items are included in your ETU package:

- ✓ One of the following sets of storage media containing the ETU software: six 5.25-inch diskettes, one reel-to-reel tape, one tape cartridge, or one CDROM.

If this item is missing, contact N Lynx Customer Service at 888-NLynx67 from 8:30 to 5:30 Central Standard Time.

## Security Considerations

When you first receive ETU, there are some functions that can only be performed by the system security officer (QSECOFR). The security level associated with each function may be changed to allow more users access to, or to limit the number of user for a particular ETU function.

Table 1-1 lists the ETU functions and the security required to use them. For more information about security with ETU, see Appendix A, Inside ETU.

**NOTE:** If a user profile contains special authority (SPCAUT) value \*ALLOBJ, all users with that profile automatically have access to the ETU power commands on the ETU Main menu. To revoke the power commands from this user profile, you must change the user profile, removing \*ALLOBJ from SPCAUT value.

Systems with security level of 10 or 20 will automatically have \*ALLOBJ in SPCAUT.

## Hardware and Software Requirements

The host and microcomputer hardware and software required by ETU are described below.

### Hardware Requirements

In order to run ETU, you must have the following hardware components:

- IBM AS/400

and one of the following:

- IBM PC, IBM PS/2, or compatible;
- Apple Mac System 8.0 to OSX.

### Software Requirements

**The AS/400 requires:**

- Operating System/400, Version 1 Release 3, or higher;
- Source Entry Utility (SEU).

**NOTE:** You must specify if your machine is a CISC or RISC version.

• **Each PC requires:**

- Windows 32-bit operating systems: Win95, Win98, SE, ME, XP, or 2000
- One of the N Lynx Technologies Inc. ES32 products: ES/PCI, ES/Remote, ES/TCP, ES/3XTwin, ES/Server, or ES/Client.

• **Each Apple Mac requires:**

- OS8 or OS9
- MacMidrange Client or TCP/Axcess by N Lynx Technologies, Inc.

**Note:** Many older products, including DOS, Unix, and Windows 3.1 will work with ETU also, but they are not supported by N Lynx Technologies. Rule of thumb, if it worked in the past, and the operating system was not upgraded, it will probably still work.

## Features of Version 5.02

This version of ETU has several new features that didn't exist in Version 5.01 (or earlier), including:

- The maximum column width is supported.
- ETU 400 supports print translations to the maximum column width allowable in OS/400. In the current version of ETU 400, the prompt for column width is 999 characters. However, the latest version of OS/400 allows a column width of only 378.
- A new translation table is included.
- ETU 400 contains a new translation table for host-to-PC file (EBCDIC to ASCII) file translations. The new translation table, ETOA3, is for fixed-length records and is suitable for some postal, IRS, and banking applications.
- The RMVETU command has been reimplemented. The RMVETU command, which gracefully removes ETU from your AS/400, has been reimplemented. See Appendix A for details.

This version of ETU is compatible with most of the earlier versions.

## Features of Version 5.041

ETU 400 RISC Version 5.041 represents an incremental maintenance release. We have modified the installation procedure for this version so that an IBM PTF does not need to be installed on the host prior to the installation of ETU 400 RISC.

### Details

With V3R7 of OS/400, the previous release of ETU 400 RISC (5.04) required IBM PTF SP36296 to be installed on the AS/400 prior to the installation of the ETU RISC CD-ROM. On OS/400 V3R7, the handling of Optical Files (OPTFILE) on the Restore Object (RSTOBJ) OS/400 command was changed. In V3R6, the parameter OPTFILE defaulted to the SAVLIB. On V3R7, this parameter needed to be explicitly specified. IBM PTF SP36296 made the Restore Objects OS/400 command work the same as in V3R6.

Version 5.04.1 of ETU 400 RISC accommodates either V3R6 or V3R7, so that the PTF is not required. This makes the installation much easier.

## Features of Version 5.5

Version 5.5 and the changes to it are as follows:

This version has been created to remove the need for registration with ETU (the Y2K changes are now redundant).

1. Changed save command to use V3R6 rather than \*PRV, so that once built ETU can be used on any version of RISC OS400.
2. Changed all RPG programs to ignore decimal data errors, to solve problems that were occurring with spreadsheet data.
3. Transfer commands with \*DIF file translation, would replace a "" (one quote) embedded in the data stream with "" (two quotes) (for the commands that were missed when V5.01 was created).
4. Corrected LOCAL/REMOTE option for when you are connected to a remote controller.
5. Changed user message to read - &bytes moved to PC, and added, &bytes moved from PC.
6. Changed the build command so that no checking is done for the version of ETU - had to keep changing it each time we built a new version.
7. There is no longer default install library - user must specify at installation.

## Features of Version 5.6

Version 5.6 is the latest release.

This release supports changes in Operating System V5R1. The issue was a problem transferring AS/400 spool files to the PC. When you select the Output queue, you could not see any of the files that were in that queue. It is downward compatible with older versions of OS/400.

### Notes to Mac Users

Most of the information in this manual is relevant to all microcomputer workstation users. However, some information is specific only to Mac users. Such information is expressed in the following format:

<b>Mac Users:</b> This is how Mac-specific information is presented.
--

### Compatibility with Non-ES32 Emulation Products

ETU is currently only supported with Nlynx Technologies **ES32** emulation. There once was a DOS Compatibility disk with a program called IBMTRAN that enabled any emulation to work. This software only works with other DOS software using serial COM ports. IBMTRAN was never ported to the Windows environment. IBMTRAN doesn't work with Windows 95 or any later Windows Operating Systems. It is not supported because the Windows Registry makes this too difficult. 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-etu400.htm> click on the link: ETU Compatibility Program for REAL DOS Only. (See ETU Compatibility Diskette: For non-ES32 Emulation Products in Chapter 2.)

### Native Mode and System/36 Mode

ETU 400 supports two sets of commands, one in the AS/400 native mode and the other in System/36 mode on the AS/400. Chapters 3 through 10 are for the native mode command set. Appendix G addresses the commands and operation for System/36 mode. System/36 mode interface does not support the new features and commands listed above.



## Part One, Getting Started

"Getting Started" is intended for all users of ETU. It describes how to use

2-1	Installing ETU
3-1	Using ETU
4-1	Learning About ETU

# INSTALLING ETU

About This Chapter

Installing ETU

    Installing the ETU Software

    Registering Your Host System

        Obtaining Your Host Registration Number

        Completing the Host Registration Procedure

Re-installing ETU

Removing ETU From Your Host System

    Installing ETU For International Users

The ETU Compatibility Diskette:

For DOS-only Non-ES32 Emulation Products

    Who Should and Can Use IBMTRAN

    Copying IBMTRAN To Your Hard Drive

    Starting ETU When Using IBMTRAN

    Exiting IBMTRAN

        Alternative Method to Start and Exit IBMTRAN

## About This Chapter

This chapter is intended for the system administrator, or other experienced host user responsible for installing the ETU software on the AS/400.

This chapter explains:

- how to install the ETU software;
- how to register ETU with N Lynx Technologies;
- how to determine if you need the IBM Compatibility Diskette in order to run ETU.

## Installing ETU

To install ETU on the AS/400, you will follow these basic steps:

- load the ETU software;
- register your copy of ETU with N Lynx Technologies
- complete the registration procedure on the AS/400.

**NOTE:** Before installing ETU, you must sign on to the AS/400 as the security officer (QSECOFR).

**NOTE:** If you install ETU to an existing library that contains the files QS36SRC or QS36PRC, the installation procedure will overwrite these files and any existing data will be lost.

### Installing the ETU Software

The ETU installation procedure copies the software from the distribution media (tape or diskette) to the AS/400. To begin the installation procedure, you must specify the library name (target library) in which ETU is to be installed. If the target library already exists (and contains a prior version of ETU), then the installation procedure updates the library with the new version of ETU. If the target library does not exist, the installation procedure creates the library. If you specify an existing library during the install procedure, any existing user files (saved request definitions, user-defined menu options) are saved to the target library. The installation procedure attempts to apply any command authorities from the existing library into the target library. In addition, you have the option of retaining your old versions of the ETU translation tables (TRANSLATE and MACTAB). The default is to load the new tables because most users do not modify the translation tables.

**NOTE:** Versions of ETU prior to 5.01 created and required the existence of the libraries PCTRAN and PCTRANHOLD. ETU Version 5.01 and all subsequent versions no longer have this requirement. If you do not plan to keep an old version of ETU on your system, you can delete library PCTRAN after installation of version 5.01 if you do not install ETU 5.01 into library PCTRAN. You can also delete library PCTRANHOLD without affecting version 5.01.

Before you start the installation procedure, do the following:

- Be sure you have at least 10MB of disk storage available on your AS/400;
- Be sure no users are accessing ETU (for existing users of ETU);
- Back up the target library if it exists prior to this installation. If a problem arises, you can restore the library to its original condition.
- *For existing ETU users:* Before you install ETU make certain that no users who are signed on to the system, including the person performing the installation, have the ETU library in their library list.

1. Sign on to the AS/400 as QSECOFR.
2. Ensure that library QTEMP is in your library list.
3. Insert ETU software in the AS/400:
  - If you are installing from diskettes, insert the first diskette into the drive on the AS/400.
  - If you are installing from tape, load the tape into the tape drive.
  - If you are installing from CDROM, insert the CDROM into the CDROM drive.
4. On the AS/400 command line, enter the RSTOBJ command interactively as follows:

For tape or diskette:

```
RSTOBJ OBJ(ETU*) SAVLIB(QTEMP) DEV(device-name) VOL(*MOUNTED)
```

where *device-name* is the name of the device in which you loaded the installation media (for example, OPT01, TAP01 or DKT01).

You should get 11 objects.

For CDROM:

```
RSTOBJ OBJ(ETU*) SAVLIB(QTEMP) DEV(OPT01) OPTFILE('/QTEMP')
```

You should get 11 objects.

For diskette users, re-insert diskette 1.

On the AS/400 command line, enter the ETU installation procedure. Type:

```
ETUINSTALL TLIBNAM(target-library) DEVNAM(device-name)  
ELIBNAM(existing-library) SVTRANL(save-translations)
```

where:

***target-library*** is the library in which ETU will reside after the install process. If the library you specify already exists, we recommend that it contain a prior version of ETU.

***device-name*** is the name of the device in which the installation media was loaded (for example, TAP01 or DKT01).

***existing-library*** is a library on your system in which a prior version of ETU exists that contains information you wish to keep in its current state. If you want to replace your current version of ETU and keep the old library name, the target-library name should be the same as the existing-library name. Specify \*NONE if there is no existing library.

***save-translations*** is for users who have modified or created ETU translation tables (the files TRANSLATE and MACTAB) in versions of ETU prior to 5.01, and wish to keep those translation tables for use with version 5.01. The default is (\*NO) which does **not** keep the

old versions, since most users do not modify these tables

If you have a version of ETU prior to 5.01 and have modified or created translation tables, you must modify the tables further for ETU 5.01. The information below assumes you are familiar with the terminology and steps required to modify the translation tables, as defined in the ETU 400 User Guide.

The host-to-micro source members for the PC (ETOA1) or the Mac (ETOA1M), or any user-defined table that was created using these members, must be modified as described below.

For ETOA1

The FROM HEX value 09 must have a TO HEX value of 09.  
The FROM HEX value 0A must have a TO HEX value of 0A.  
The FROM HEX value 0D must have a TO HEX value of 0D.

For ETOA1M

The FROM HEX value 09 must have a TO HEX value of 09.  
The FROM HEX value 0A must have a TO HEX value of 0D.  
The \*EOR record must be changed from 0D0A to 0D.  
The \*EOF record must be changed from 1A to blank since the Mac uses no end-of-file character.

After you change the source, be sure to run the EDITTABLE command.

## Registering Your Host System

ETU 400 is licensed to be installed on a single host system, and may be run on only that system. After you install the ETU software on your host, you are granted a 30-day evaluation period.

Older versions of ETU (CISC) require specific host registration. If you have purchased an older version of ETU, you must obtain a host registration number from N Lynx Technologies during this 30-day period to allow unlimited use of the software on your host system.

<p><b>NOTE:</b> Once you receive your ETU host registration number from N Lynx Technologies, you may not return ETU for a refund.</p>
---

Newer versions of ETU do not require registration. We will require a serial number prior to providing Technical Support.

## Using ETU During the Evaluation Period

You do not need a host registration number while you are using ETU in the evaluation period. However, until you register ETU with N Lynx Technologies, you will see a reminder prompt whenever you issue any ETU transfer command. Do not press Enter at the prompt.

## Obtaining Your Host Registration Number

1. Have the following information ready before you request your host registration number.
  - ✓ The ETU serial number located on your ETU software media. (For most software media, the serial number is six digits beginning with 2. For 8mm tape, the serial number is eight digits beginning with 5.)
  - ✓ Your host serial number. Display your host serial number by issuing the following command from the AS/400 command line:  
**dspsysval sysval(qsrlnbr)**
  - ✓ The AS/400 model you have.
  - ✓ The ETU version (for example, 5.02)
  - ✓ The name of your company, a person to contact, and a phone number.
  - ✓ FAX or phone the above information to N Lynx Technologies between the hours of 8:30 a.m. and 5:30 p.m., Central Time, Monday through Friday.

U.S. FAX: 512-301-8030  
U.S. phone: 888-NLynx67

European FAX: 44-161-928-7015  
European phone: 44-161-928-7014

We will phone or FAX your host registration number to you.

## Entering the Registration Code

Once you receive your host registration number, you can complete the registration procedure.

1. Add the library containing ETU to your AS/400 library list. At the AS/400 command line, type:

**addlible *library-name***

where *library-name* is the library containing ETU.

2. Enter the OWNERID command at the AS/400 command line:

**ownerid**

You will be prompted for the host registration number.

3. Enter your host registration number (type any alphabetic characters in UPPER CASE).
4. Press ENTER to exit the OWNERID screen.

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

## Re-installing ETU

If you are re-installing ETU in order to recover from a system crash or some other system error, re-install ETU following the instructions in Installing the ETU Software on page 2-3.



If you are updating an existing version of ETU, sign on as the system security officer (QSECOFR), back up your ETU library using the SAVLIB command, and then re-install ETU following the instructions in Installing the ETU Software on page 2-3.

## **Removing ETU From Your Host System**

ETU includes a utility that allows for easy removal of the ETU programs from your host system. See Appendix A for details on removing ETU.

## **Installing ETU For International Users**

If you are running ETU on a host system that supports a language other than U.S. English, you may need to install the foreign language translation table appropriate for your language. To install foreign language tables, see Appendix H, International Translation Tables.

## **Technical Support**

Technical Support is provided for up to one year from the date of purchase. Serial number is required. Technical Support by telephone is from 8:30 - 5:30 CST. Toll-free 1-888-NLynx67 (Technical Support Hotline)

We will provide answers and support by email for product that is not under warranty if you use the Technical Support Contact Form at <http://www.nlynx.com/html/contacttechnicalsupport.htm>

Technical Bulletins for ETU are located at <http://www.nlynx.com/html/tb-etu.htm>

## The ETU Compatibility Diskette: For Non-ES32 Emulation Products

When ETU was originally sold, there was a compatibility disk that shipped with the product to allow **DOS** products to run with it. Very few PC run in DOS mode anymore, so that is no longer included with this product, however it is available on our website at <http://www.nlynx.com/html/tb-etu36.htm>. To use this program, called IBMTRAN, you would install it on the IBM PC from which you run ETU. Each time you run ETU, you would load IBMTRAN into the PC's memory.

### Who Could Use IBMTRAN?

You may need to use IBMTRAN if your emulation software is **not** one of the following Andrew products: 3XTwin, 3XPlus, 3XPlus-X.25, 3XMate, 3XTwin-Laptop, 3XMate-Laptop, Multi-Twin, or MacTwin. Additionally products in the Nlynx Systems ES/95 family do not need it, and products in the Nlynx Technologies ES32 family do not need it.

**NOTE:** IBMTRAN runs only under DOS and isn't intended for use with Windows, OS/2, UNIX, or the Apple Mac.

Some emulation products, such as protocol converters, have their own ETU interface programs. If your emulation package includes a program specified as an interface to ETU, you should use that program instead of IBMTRAN.EXE.

If you attempt an ETU file transfer and you receive one of the following messages, you may need to use IBMTRAN:

**Device not a PC or emulator loaded incorrectly.**

**Attached workstation is not a PC.**

### Copying IBMTRAN To Your Hard Drive

If you decide you need the compatibility program, copy it from the ETU Compatibility diskette (enclosed with ETU) to the hard disk subdirectory (or, if applicable, to the floppy diskette) that contains the emulation programs.

1. On your PC, make the directory with your emulation programs the current directory.
2. Insert the IBMTRAN diskette in the diskette drive, and copy it to the hard drive.

If your floppy drive is a:, enter:

**copy a:\*.\***

Some emulation programs have a separate Application Program Interface (API) program. If your emulator is in this category, make sure that API is running after you load your emulation program and before you run ETU, or it will not be compatible with ETU.

If your emulation program is running, and IBMTRAN displays the message

"5250 Emulation not loaded, verify that your API is loaded."

If your API is loaded and this message persists, contact your emulation manufacturer to be sure you have the latest version.

The documentation included with your emulation software should provide you with details about API.

**NOTE:** If you configure your emulator and API the way you would to run PC Support/36, then IBMTRAN should work.

### **Starting ETU When Using IBMTRAN**

Once you've copied IBMTRAN to your PC, follow these instructions to load IBMTRAN into your PC's memory.

1. Load the emulator and establish a session with the host.
2. Hotkey to DOS (press ALT-ESC).
3. Run the IBMTRAN.EXE program on the PC. To do this, make current the subdirectory containing the emulation software. Then type:

**ibmtran**

4. After the program loads, it automatically hotkeys back to emulation. You can now run any ETU command.

### **Exiting IBMTRAN**

After you load IBMTRAN on your PC, it remains in memory until you unload it. To exit IBMTRAN:

1. Hotkey to DOS (press ALT-ESC). An intermediate screen displays, which contains the following message across the bottom of the screen:

**ETU is now active. ALT-ESC to return to emulation, ALT-BREAK to return to DOS.**

2. Press ALT-BREAK to unload IBMTRAN.  
This will unload IBMTRAN from memory, and leave you at the DOS prompt. You can hotkey from DOS back to the host session and continue working in emulation. Remember to load IBMTRAN again before you attempt another ETU file transfer.

### **Alternative Method to Start and Exit IBMTRAN**

You can also load IBMTRAN by typing **ibmtran**. This variation returns to DOS, thus bypassing the intermediate screen described above. This may be helpful for use in batch files where you want to perform a file transfer and then exit out of IBMTRAN.

# USING ETU

About This Chapter

Starting ETU

    A Note About Authorization

The ETU Main Menu

    The ETU Utilities Menu

The ETU Commands

    Main Menu: File Transfer Commands

    Main Menu: ETU Power Commands

    Utilities Menu: Microcomputer Commands

    Utilities Menu: File Translation Commands

    Utilities Menu: Translation Table Commands

Running ETU Commands

    Using the Menu

    Entering Commands on the Command Line

    Using the Command Prompts

    Using the Mac Directory Dialogue Boxes

    Recalling ETU Commands

        Request Definitions

Types of Processing

    Interactive ETU Commands

    Batch ETU Commands

About Microcomputer Files

    File Formats

        PC-DOS

        Apple Mac

## About This Chapter

This chapter is intended for beginning users of ETU. It covers the basics of running ETU in native mode on the AS/400. In this chapter, you will learn:

- how to start ETU;
- how to use the ETU menu;
- how to enter ETU commands;
- about using ETU with PC or Apple Mac files.

Once you have read this chapter, continue with Chapter 4, Learning About ETU.

## Starting ETU

You can begin using ETU once it is installed on your host system.

To start ETU:

1. Load your 5250 emulation software on your microcomputer. If necessary, load the ETU compatibility program, IBMTRAN, on your PC. See Chapter 2, *Installing ETU*, for information about IBMTRAN.
2. Enter your AS/400 user I.D. and password at the sign-on screen.
3. Display the ETU menu by entering the following command on the AS/400 command line: `wsmenus`

**NOTE:** The AS/400 library containing ETU must be in your library list.

When the ETU menu displays, you can execute any ETU command, provided you have proper authorization.

### A Note About User Authorization

Not all users will be able to use all commands. The commands you are allowed to use depend on the authorization you are granted by the system administrator. If you don't have the authority to use a certain command, it won't display on the ETU menu, nor will you be able to run the command from the command line. See Appendix A, *Inside ETU*, for more information about user authorization with ETU.

## The ETU Main Menu

The ETU commands are split between two menus, the Main menu and the Utilities menu. The Main menu, shown in Figure 3-1, displays when you issue the command WSMENUS. It contains an option for each file transfer command. In addition, it also contains an option that displays the ETU Utilities menu, and an option that signs you off the AS/400.

The right side of the Main menu is reserved for user-defined functions, which are site-written commands that the system administrator can assign to the ETU Main menu. A user-defined function might be a menu option that, when selected, starts a non-ETU related program. Your system administrator can add any user function to the ETU Main menu. If no user functions have been added, the right side of the menu is blank, as it appears below in Figure 3-1.

For information on adding user-defined functions to the Main menu, see the command EDTUDF (edit user-defined functions) in the Chapter 10, ETU Command Reference.

**NOTE:** The menu options 7 through 10 (for the ETU power commands) display on your menu only if you have been granted authority to them, or if you have special authorities granting access to all objects.

Figure 3-1 The ETU Main Menu

```
Menu - WSMENUS          Emulator Transfer Utility AS/400          NLRISC
NLRISCLIFF             COPYRIGHT (C) 1998 - Nlynx Systems

    ETU Functions
    1. Host file to PC file
    2. Host program to PC file
    3. Host spool file to PC prt file
    4. PC file to Host file
    5. PC file to Host program
    6. PC prt file to Host spool file
    7. Power Host file to PC file
    8. Power PC file to Host file
    9. PC from PC
   10. PC to PC
   20. ETU Utilities Menu
   90. SIGNOFF

OPTION OR COMMAND
====>

F3= Exit                F4= Prompt                F9= Retrieve
F6= DSPMSG
```

## The ETU Utilities Menu

The ETU utilities menu provides access to the ETU commands that aren't used for transferring files. There are three groups of commands on the Utilities menu. They are the PC functions commands, the translation commands, and the translation table commands. These commands are described later in this chapter.

To display the ETU Utilities menu, select option 20 from the ETU Main menu.

Figure 3-2 The ETU Utilities Menu

```
Menu - WSMENUS          Emulator Transfer Utility AS/400          NLRISC
NLRISCLIFF             COPYRIGHT (C) 1998 Nlynx Systems
                        ETU Utilities Menu

      PC Functions
1. Allocate NEW PC file
2. Rename PC file
3. Delete PC file
4. Test for existence of PC file

      Translation Functions
8. Host file to PC file
9. Host spool file to PC prt file
10. PC file to Host file
11. PC prt file to Host spool file

OPTION OR COMMAND
====>

F3= Exit                F4= Prompt                F9= Retrieve
F6= DSPMSG              F12= Cancel
```



## The ETU Commands

There are 20 ETU commands that work with files. These commands can be divided into five groups, each providing a different function. The ETU commands are grouped below according to the menu on which they appear, and their functions.

### Main Menu: File Transfer Commands

These commands transfer files between a microcomputer and the host. They are discussed in Chapter 5, Transferring Files.

Option 1.	FROM3XB	Option 4.	TO3XB
Option 2.	PGMFROM3XB	Option 5.	PGMTO3XB
Option 3.	PRTFROM3X	Option 6.	PRTTO3X

### Main Menu: ETU Power Commands

These commands offer advanced features to experienced users of ETU. They are discussed in Chapter 9, Advanced ETU Features.

Option 1.	PWRFROM3XB	Option 3.	PCFROMPC
Option 2.	PWRTO3XB	Option 4.	PCTOPC

### Utilities Menu: Microcomputer Commands

These commands allow you to work with micro files while you are working on the host in emulation. They are discussed in Chapter 7, Working with PC Functions.

Option 1.	ALOCATB	Option 3.	DELETEDB
Option 2.	RENAMEB	Option 4.	TESTB

### Utilities Menu: File Translation Commands

These commands translate data so that both the host and the microcomputer can use it. They are discussed in Chapter 6, Translating Files.

Option 1.	XLATETO	Option 3.	XLATEFROM
Option 2.	PRTXTO3XB	Option 4.	PRTXFRMPC

### Utilities Menu: Translation Table Commands

These commands allow advanced users to change the translation tables supplied with ETU. They are discussed in Chapter 8, Modifying Translation Tables.

Option 1.	STRSEU	Option 2.	EDITTABLE
-----------	--------	-----------	-----------

## Running ETU Commands

With ETU, there are two ways to run commands: using the ETU menu, or entering ETU commands on the command line. Both methods produce the same result; that is, a prompt screen displays on which you provide information like the name of the file you want to transfer, and where you want to transfer the file. ETU menus have an option field or a command line, but not both, depending on your authority.

### Using the Menu

Once you display the menu using the WSMENUS command, enter the menu option number that corresponds to the command you want to run. ETU displays a prompt screen on which you enter the information necessary to carry out, or execute, the command. The ETU commands are listed later in this chapter.

### Entering Commands on the Command Line

Each ETU function has a command name that is recognized by the AS/400. To execute any ETU command, enter the command name on the AS/400 command line, and prompt (press F4) for the information about that command. For example, to transfer a file from the workstation to the host using the T03XB command, enter T03XB on the command line and press F4. This displays the T03XB prompt screen (see Figure 3-3 below) on which you enter the information about the transfer.

The prompt screen that displays when you press F4 is identical to the one that displays when you initiate the command from the menu.

**NOTE:** Depending on your authority level, the ETU menu will display either an option field or a command line.

**Figure 3-3 The T03XB prompt screen**

```
Transfer to 3X: (T03XB)

Type choices, press Enter.

File name: . . . . . Name
Library name: . . . . . *LIBL Name, *LIBL
Member name: . . . . . *FILE Name, *FILE
Record length (0=existing): . . 0-9989
File type: . . . . . *DATA *DATA, *SRC, *SAVE, *SAVF
Add seq/date field?: . . . . . *NO *YES, *NO
Type of translate?: . . . . . *TEXT *NO, *TEXT, *DIF, *BASICS...
Translation table file name: . . TRANSLATE Name, TRANSLATE, MACTAB
Library name: . . . . . *LIBL Name, *LIBL
Member name: . . . . . ATOE1 Name
Truncate text: . . . . . *NO *YES, *NO
DDS or F & I file name: . . . . *NONE Name, *NONE, *DDS
Library name: . . . . . *LIBL Name, *LIBL
Member name: . . . . . *FILE Name, *FILE, *FIRST, *LAST
Qualified file name: . . . . .

More...

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

## Using the Command Prompts

After you select the ETU command that you want to run, either by choosing an option from the menu or by entering the name of the command on the command line and pressing F4, a prompt screen displays. This screen lists the parameters associated with the command. A parameter is a placeholder for a single piece of information, or value, that you must provide to ETU to tell ETU what to do and how to do it.

When the prompt screen displays, enter a value for each parameter that applies to your transfer operation. For instance, when you transfer a host file to the microcomputer, you must provide, among other things, the name of the host file. You enter the name of the host file in the FROMFILE parameter. Thus, the host file name is the value to be entered in that parameter.

Some parameters contain preset values, called defaults. If you do not need to use a particular parameter, leave it at its default value. To view all the possible values for a particular parameter, enter a question mark (?) in the first position of the parameter, press FIELD EXIT, and then press ENTER. If there is a plus sign (+) next to the parameter keyword, the parameter can be extended for additional values. To enter the additional values, place the cursor next to + for more, enter a plus sign, and press ENTER.

When you have filled out all the required parameters, press the ENTER key. If the command contains more than one screen of parameters (which some ETU commands do), press PAGE UP to display subsequent screens. To start the command, press ENTER.

## Using Mac Directory Dialog Boxes

Mac directory dialog boxes are available for some ETU commands. A directory dialog box provides an alternative to entering a value into a menu's micro file name prompt.

You can specify the location of the Mac file by entering the file name at the QFNAME1 prompt, or you can display a dialog box. To display a dialog box, enter an asterisk (\*) in the first position in the QFNAME1 prompt, fill out all other parameters required for the transfer operation, and press ENTER.

**NOTE:** The directory dialog boxes are unavailable for the DELETEDB and RENAMEDB procedures.

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 receive an ETU user message giving you the option to CANCEL or GO. Select CANCEL to return to the WSMENUS options.

## **Recalling ETU Commands**

If you run the same ETU command more than once, you can avoid re-keying repeated values by using the recall function. The recall function displays the most recently run command on the ETU command line.

To recall a command, press F9. You can then change whatever information you want by typing over it. You can cycle through and recall any command you entered since you accessed the ETU menu (WSMENUS) by repeatedly pressing F9. Upon exiting ETU, all references to previously-entered commands is lost to future executions of WSMENUS.

Another way to re-use previously-entered commands is by saving and re-using request definitions.

## **Request Definitions**

Some ETU commands (FROM3XB, PCFROMPC, PCTOPC, PWRFROM3XB, PWRT03XB, and TO3XB) support savable request definitions. A request definition is the information you provide to ETU to run a command. For example, if you transfer a file from the host to the micro using the FROM3XB command, the request definition is the summation of the values you entered for each parameter.

The commands that support savable request definitions (listed above) contain a parameter called SAVRQS (save request). This parameter allows you to save the command request for the transfer operation that you are currently performing. To save the definition of a command request, enter \*YES as the value in the SAVRQS parameter. Then supply a name for the request. The request name may be up to ten characters long.

To replay any request definition you have previously saved, use the ETU command VWSAVRQS (view save request). The VWSAVRQS command is covered in Chapter 9, Advanced ETU Features and Chapter 10, ETU Command Reference.

<p><b>NOTE:</b> When you save a PWRFROM3XB request definition, note that special values shouldn't be used in the Comparison Value prompt in the Record Selection parameter. See PWRFROM3XB command in Chapter 10, Command Reference for more details on the special values.</p>
---

## Types of Processing

Any work that is processed by the AS/400 is called a job. Two important types of jobs are interactive jobs and batch jobs. All ETU commands can be run interactively. Some can be run in batch mode, as well.

### Interactive ETU Commands

Interactive jobs begin processing the moment you execute them, usually when you press the ENTER key. The microcomputer from which you execute the job is input inhibited or tied up until the interactive job is completed. During this time you can't use the host session from which you have initiated ETU.

The following commands are interactive and can be run only from a microcomputer emulating a 5250: ALOCATB, DELETEDB, TO3XB, PGMT03XB, PGMFROM3XB, RENAMEB, TESTB, FROM3XB, PRTFROM3X, PRTTO3X, EDITTABLE, and STRSEU.

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

The power commands can be initiated from any AS/400 display. The source or target device specified in the transfer must be a microcomputer running 5250 emulation with the AS/400 sign-on screen displayed.

### Batch ETU Commands

Batch jobs are submitted to a job queue where they are processed along with any other jobs on the queue. Batch jobs do not tie up your microcomputer.

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

The ETU power commands can also be submitted to batch by using SBMJOB or the ETU utility SBMBATXFER (submit batch transfer). The power commands are PWRDELETEDB, PCFROMPC, PCTOPC, PWRFROM3XB, PWRRENAMEB, and PWERTO3XB.

## About Microcomputer Files

ETU works with several types of microcomputers, including IBM PCs (and compatibles) and Apple Macs. Throughout this guide, these computers are referred to as micros. When you perform an ETU command that works with a PC or Mac file, you will be asked to provide the micro file name. The micro file names can contain up to 80 characters, including the file path.

### File Formats

Because of the various microcomputers supported by ETU, the micro file name must follow the format required by the operating system of the attached microcomputer. Descriptions of the file name formats for PC-DOS and the Apple Mac are listed below. Refer to the user guide for your microcomputer's operating system for additional information.

#### PC-DOS

Format: drive:\pathname\micro\_file\_name.ext

Example: C:\SUBDIR1\EXAMPLE.TXT

where:

```
drive is C:
pathname is SUBDIR1
micro_file_name is EXAMPLE.TXT
```

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

#### Apple Mac

Format: disk:pathname:micro\_file\_name

Example: APPLEDRIVE:FOLDER1:WORKSHEET

where:

```
disk is APPLEDRIVE
pathname is FOLDER1
micro_file_name is WORKSHEET
```

This example applies to Mac files transferred to one destination. See Chapter 9, *Advanced ETU Features*, for information about transferring files to or from multiple Macs.

**NOTE:** If you enter an ETU command on the command line or in a CL procedure, you must place single quote marks (') around the microcomputer's path and file name. For example, 'C:\SUBDIR1\EXAMPLE.TXT'

# LEARNING ABOUT ETU

About This Chapter

What is ETU?

    Data Transfer

    Data Translation

    Other ETU Functions

About Transferring Files

    Micro and Host File Names

    Transferring Data to Host Files

        Record Length

        Host File Type

        Data Description Specifications

About Translating Files

    How Data is Stored

Using Translation Tables

    ETU Translation Table Members

    Using Translation Tables With the Mac

Using Microcomputer Data Formats

## About This Chapter

This chapter is a general overview of ETU and is intended for beginning users. It explains how data is stored on the host and the microcomputer, and how ETU makes it possible to transfer data between the two systems.

The information you provide when you transfer and translate files using ETU is also explained in this chapter.



## What is ETU?

ETU has two main purposes: data transfer and data translation. Data transfer entails the actual movement of data between the host and a microcomputer. Data translation changes the way the data is represented or stored so that the same data can be used by the two different systems. ETU provides you with file transfer commands and file translation commands.

### Data Transfer

With the file transfer commands, you can transfer and translate the file, or you can specify that translations not occur. The transfer commands are discussed in Chapter 5, Transferring Files.

### Data Translation

If you don't translate a file with a transfer command, you can translate it later using one of the ETU translation commands. There is a translation command that corresponds to each transfer command. (The ETU commands used to transfer host programs, `PGMT03XB` and `PGMFROM3XB`, have no corresponding translation commands. This is because host programs are never translated to a micro format.)

One reason you might use a translation command is if you are transferring a file to or from a remote micro via a modem. Rather than have data translation take place over the phone line, the data can be translated more efficiently on the host system.

For example, if you are transferring a file from the host to a remote micro, you can first translate the data from any micro or terminal attached to the host using one of the file translation commands. When the translation is complete, you can then transfer the file to the remote micro with the appropriate file transfer command. This reduces the amount of expensive telephone line time used. The translation commands are discussed in Chapter 6, Translating Files.

### Other ETU Functions

In addition to the transfer and translation commands, ETU includes a set of commands that allow you to work with PC files. With these commands, you can perform PC functions, such as creating and deleting PC files, while you are working in emulation. These commands are described in Chapter 7, Working With PC Functions.

This guide also discusses the ETU commands for advanced users. These commands allow experienced users to modify the translation tables included with ETU, and to perform other advanced functions such as transferring files to multiple devices, transferring selected records, and using entry and exit programs. To learn more about these commands, see Chapter 8, Modifying Translation Tables, and Chapter 9, Advanced ETU Features.

## About Transferring Files

ETU allows you to transfer files and programs between a host system and any microcomputer connected by emulation software to the host. To transfer a file from one system to another, you must provide certain information such as the name of the file you are transferring, the type of file it is, and the name of file to which you are transferring it.

For example, if you are transferring a file from a PC to the host, you must provide the name of the PC file and the name of the host file. If the file that is to receive the data does not exist, ETU will create the file. The information common to the file transfer commands is explained below. For more information about specific transfer commands, see Chapter 5, Transferring Files.

### Micro and Host File Names

ETU requires you to supply the names of the host file and the micro file you want to transfer or translate. You should be familiar with the following file name parameters:

- |                            |   |
|----------------------------|---|
| <b>Qualified file name</b> | The name of the micro file you are transferring data to or from. The qualified file name can be up to 80 characters long including the file path. See Chapter 3, Using ETU, for more information on micro files.  |
| <b>Host file name</b>      | The name of the host file that you are transferring data to or from. Host file names can be up to ten characters long. You must include the name of the host file for all file transfers.   |
| <b>Host library name</b>   | The library containing the host file. You may include the name of the library or use the default, *LIBL, which searches all libraries in your library list for the host file.   |
| <b>Host member name</b>    | The name of a specific member in a host physical file, if the physical file contains more than one member. You can transfer one specific member by including the member name, or you can use the default, *FILE, which transfers the member with the same name as the host file name. |

## Transferring Data to Host Files

When you transfer data to the host, you can transfer it to an existing file, or to a new file, which ETU creates.

To transfer data to an existing file, supply the name of the existing file, the library it resides in, and the member name. The data within the file is defined using the existing file's definition.

To transfer data to a new file, supply the new file's name, the library in which it will reside, the member name, and the file type. In addition, you must define the file that ETU creates. There are two methods for defining files. For externally described files, you must define the file by creating Data Description Specifications (DDS). For non-externally described files, such as text files, define the file by providing a record length.

## Record Length

For new host files that are not externally described, you must enter the file's record length, in bytes, at the RECL prompt. For existing host files, and new files defined with DDS specifications, the record length should be zero (the default).

## Host File Type

When you transfer a file to the host from the micro with the T03XB or PWRT03XB commands, ETU will create the host file if the file doesn't already exist. If ETU creates the host file, you must specify the type of host file it is to be. ETU supports four host file types. They are:

- \*DATA** A host physical file. The most common host file type. \*DATA is the default file type. To create a \*DATA file, you must specify a record length or define the file using Data Description Specifications.
- \*SAVE** A \*SAVE file permits micro files to be transferred to the host without being translated. This allows you to use the host as an off-line storage site for your micro files. If you use \*SAVE, you must specify record length 1024 and TRANSLATE (\*NONE). To transfer a \*SAVE file back to the micro, use the FROM3XB command with SAVTYPE(\*YES), TRANSLATE(\*NONE), and RECL(1024).
- \*SRC** A source physical file is a file that can be read by the AS/400 Source Entry Utility (SEU).
- \*SAVF** A file type that provides a means of storing AS/400 objects on line. To use \*SAVF, RECL (record length) must be 528 and TRANSLATE (\*NONE).

## Data Description Specifications

When you create a physical file (\*DATA) on the AS/400, you can define the file using Data Description Specifications (DDS). The DDS file definition tells the computer many things about the file, including where one field begins and ends, and whether a field is numeric or alphabetic.

It's not necessary for each physical file to have a unique set of DDS specifications. In fact, if the format is the same for two different files, you can define them both with the same DDS specifications (be sure to use different file names when creating the physical files).

The DDS definition is stored as a member within a source physical file. When you are transferring data to a new host file, you can specify the source file and member containing the DDS that define the file. Specify the file containing the DDS in the parameter called FLDDEFFILE.

**NOTE:** If, when transferring data to a new host file, you specify both a record length (RECL) and a DDS member, the record length takes precedence. As a consequence, the DDS member you specify will not be used in the transfer and the data may not appear on the host as you expected.

Appendix C in this guide explains how to create DDS on the AS/400.

## About Translating Files

You must usually translate data when you transfer it between a microcomputer and the host. That's because the host and the microcomputer have very different data storage methods. The remainder of this chapter explains the differences between host and micro data, and the two types of translations ETU performs to overcome those differences.

### How Data is Stored

The way data is stored on the AS/400 is considerably different from the way it is stored on the micro.

On the AS/400:

- Files are stored in fixed-length records;
- Field definitions are usually maintained external to the file;
- Data is stored in an EBCDIC character set.

On the PC:

- Files are stored in fixed or variable-length records;
- 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 ETU performs two types of translations. It translates data between ASCII and EBCDIC, and it also reformats the data into one of the various microcomputer file formats. To perform the ASCII-to-EBCDIC translation, ETU uses translation tables, which are explained on the following page. The different microcomputer file formats are explained on page 4-11.

## Using Translation Tables

ASCII and EBCDIC are standard character sets. When you transfer a file between a microcomputer and the host, the file must be translated from one character set to the other. (Unless, that is, you are only storing the file and not planning to manipulate or use the file in any way on the other system. In that case, you can transfer files without any translation taking place by specifying \*NONE as the value for the TRANSLATE parameter.)

When you translate a file, you must specify a translation table. The translation table tells ETU whether

- the file is being transferred from the host to the micro, or from the micro to the host;
- the file has a fixed- or variable-length record;
- the file is a print file or a data file;
- the micro involved is a DOS-based PC or a Mac.

ETU comes with seven translation tables; four for PC-DOS translations and three for Mac translations. The translation tables are actually members within two files. All translation tables, for both PC-DOS and Mac, are stored in the file TRANSLATE. For convenience, the Mac members alone reside in the file MACTAB. The member names are described below.

When translating a file, specify the translation table member that is appropriate for your situation. For example, if you are translating a file from a Mac to the host, use translation table ATOE1 in file MACTAB. For PC to host translations, use the translation table ATOE1 in file TRANSLATE.

To specify a translation table, enter the appropriate translation file name (either TRANSLATE or MACTAB) and the translation member name (for example, ETOA1) at the TRANSFILE prompt.

If these standard tables do not meet the needs of a specific application, you may change them. For more information about creating and modifying translation tables, see Chapter 8.

## ETU Translation Table Members

The ETU translation tables member names and their specific functions in the TRANSLATE file are listed below.

- ETOA1** The EBCDIC-to-ASCII PC-DOS character translation member that produces a variable-length record, with carriage return and line feed added after the last byte of data in each record. Trailing blanks are stripped.
- ETOA2** The EBCDIC-to-ASCII PC-DOS character translation member that produces a basic random file on the PC, with no carriage return line feed. The PC application is expected to read for a fixed record length.
- ETOA3** The EBCDIC-to-ASCII PC-DOS character translation member that produces a fixed-length record, with carriage return and line feed added at the end of each record. Trailing blanks are not stripped. May be suitable for postal, IRS, or banking applications.
- ATOE1** The ASCII-to-EBCDIC PC-DOS fixed- and variable-length translation member.
- ATOE3** The ASCII-to-EBCDIC PC-DOS print translation member.

**Mac Users:** The Mac-specific translation tables that reside in the MACTAB file are described below. Note that they have the same name as their TRANSLATE counterparts except for the **M** on the end.

- ETOA1M** The EBCDIC-to-ASCII Mac character translation member that produces a variable-length record, with carriage return and line feed added after the last byte of data in each record. Trailing blanks are stripped.
- ETOA2M** The EBCDIC-to-ASCII Mac character translation member that produces a basic random file on the PC, with no carriage return line feed. The PC application is expected to read for a fixed record length.
- ATOE1M** The ASCII-to-EBCDIC Mac fixed- and variable-length translation member.

## Using Translation Tables with the Mac

All ETU commands in which data translation occurs include the TRANFILE parameter. The default value for this parameter is TRANSLATE, which exists for translating PC data - not Mac data. **If you use a Mac you must change the default value to MACTAB.** (If you create a different translation table, then you can specify the new table.)

As an example, the screen in Figure 4-1 displays when you select WSMENUS option #1, FROM3XB. This command transfers data from the host to the micro. Note that the preset translation table file name is TRANSLATE. To correctly translate the data for the Mac, you *must* change the value to MACTAB.

Figure 4-1 The FROM3XB prompt screen

```

Transfer from 3X: (FROM3XB)

Type choices, press Enter.

File name: . . . . .
  Library name: . . . . .      *LIBL      Name
Member name: . . . . .      *FILE      Name, *LIBL
*SAVE file type?: . . . . .  *NO      Name, *FILE, *FIRST, *LAST...
Strip seq/date field?: . . . *YES, *NO
Type of translate?: . . . . . *NO      *YES, *NO
Translation table file name: . *NO, *TEXT, *DIF, *BASICS...
  Library name: . . . . .      *LIBL      Name, TRANSLATE, MACTAB
Member name: . . . . .      ETOA1      Name
Record length (0=existing): . . *FILE      0-9989, *FILE
DDS or F & I file name: . . . *NONE      Name, *NONE, *DDS
  Library name: . . . . .      *LIBL      Name, *LIBL
Member name: . . . . .      *FILE      Name, *FILE, *FIRST, *LAST
Qualified file name: . . . . .

More...

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

Table 4-1 lists the ETU commands that require a translation table file name value. It also lists the commands' corresponding WSMENUS option numbers. For these commands, Mac users *must* change the TRANSLATE parameter to MACTAB.

Table 4-1 ETU Commands that require a translation table file name

<i>ETU Command</i>	<i>ETU Utilities Menu Option</i>
EDITTABLE	20
PRTXFRMPC	11
PRTXTOPC	9
XLATEFROM	10

Table 4-2 lists the ETU commands that use a translation table file name but don't *require* the translation. If you do translate the file using these commands, you must specify a translation table. For these commands, Mac users that use a translation table with these commands *must* change the TRANSLATE parameter to MACTAB.



Table 4-2 ETU Commands in which the translation table file name is optional

<i>ETU Command</i>	<i>ETU Main Menu Option</i>
FROM3XB	1
PRTTOPC	3
TO3XB	4
PRTFRMPC	6
PWRFROM3XB	7
PWRTO3XB	8

## Using Microcomputer Data Formats

In addition to translating files from ASCII to EBCDIC, ETU formats the data so that it can be used in the various types of microcomputer applications such as spreadsheets, word processors, and database programs. Each type of application stores data in its own format. ETU provides four translation types that enable the host data to be formatted for the type of application in which you will use it.

As an example, you may want to transfer a host file to the microcomputer and use the file in a database program. By using the \*TAB translation type (used with database applications) when you transfer the file to the micro, the file can be imported into a database application.

You must also specify a translation type when you are translating files from the micro to the host. ETU strips off the formatting created by the microcomputer application, which allows the data to be used on the host.

To specify a translation type, enter the appropriate value at the TRANSLATE parameter in any of the file transfer commands. To transfer a file between the host and a micro with no translation, enter \*NONE at the TRANSLATE parameter.

The four micro formats supported by ETU are explained below.

### **ASCII Text format (\*TEXT)**

ASCII Text is a character-oriented format. Use \*TEXT with files created with any kind of word processing program or text editor.

### **Basic Sequential format (\*BASICS)**

BASIC Sequential format is a standard method of storing data on the PC or Mac. These files are field-oriented and are created with database programs. All fields are separated by commas, and alphabetic fields are surrounded by quotation marks (") to distinguish them from numeric fields. Data in a \*BASICS format looks similar to that shown below:

Table 4-3 Example of \*BASICS data file

9.00,"LO111","10119","BLACK","DOG",9.00,"890101"
9.00,"LO118","10119","RED","DOG",8.00,"890301"
9.00,"LO118","10221","BLACK","BIRD",6.00,"890401"
9.00,"LO118","10222","BLUE","CAT",6.00,"890201"
8.00,"LO118","10222","BLACK","HORSE",7.0,"890202"
8.00,"LO118","10224","GREEN","FISH",7.0,"890202"

**Tab format (\*TAB)**

TAB format is the same as BASIC Sequential, except that tab characters, rather than commas, separate the fields. Data in a \*TAB format looks similar to that shown below:

Table 4-4 Example of a \*TAB data file

9.00 "LO111" "10119" "BLACK" "DOG" 9.00 "890101"
9.00 "LO118" "10119" "RED" "DOG" 8.00 "890301"
9.00 "LO118" "10221" "BLACK" "BIRD" 6.00 "890401"
9.00 "LO118" "10222" "BLUE" "CAT" 6.00 "890201"
8.00 "LO118" "10222" "BLACK" "HORSE" 7.0 "890202"
8.00 "LO118" "10224" "GREEN" "FISH" 7.0 "890202"

**DIF format (\*DIF)**

DIF format is a standard method of storing PC files created using a spreadsheet program. The DIF format is based on vectors and tuples (fields and records). Data in a \*DIF file looks similar to that shown below:

Table 4-5 Example of a \*DIF data file

TABLE
0,1
"."
VECTORS
0,0006
"."
TUPLES
0,000000012
"."
DATA0,0
"."
-1,0

**NOTE:** Most micro applications provide utilities that allow the data to be imported to or exported from \*BASICS, \*TAB, \*DIF, or \*TEXT. Check the documentation supplied with your PC software for more information about translating data to one of these formats.

In order to support the above-named formats, any file that ETU translates must be defined with the same field definitions as the file being transferred. The fields may be defined through the normal Data Description Specifications (DDS) member associated with the specified file name. See the IBM manual AS/400 Programming: Data Description Specifications Reference, SC21-9620, for information on creating DDS specifications.

## *Part Two*

# Running ETU

"Running ETU" is intended for all users of ETU. It introduces each ETU command and describes how to use the commands. An example of how to use each command is also provided.

### **Transferring Files**

### **Translating Files**

### **Working with PC Functions**

### **Modifying Translation Tables**

### **Advanced ETU Features**

# TRANSFERRING FILES

## About This Chapter

### Transferring Data Files

Transferring Data Files From Host to a Microcomputer:  
The FROM3XB Command

FROM3XB Example

Transferring Data Files Microcomputer to a Host:  
The TO3XB Command

TO3XB Example

### Transferring Print Files

Transferring a Host Spool File to a PC Print File:  
The PRTFROM3X Command

PRTFROM3X Example

Transferring a PC Print File to a Host:  
The PRTTO3X Command

PRTTO3X Example

### Transferring Host Programs

Transferring a Host Programs to a Microcomputer:  
The PGMFROM3X Command

PGMFROM3X Example

Transferring a PC Program to a Host Program:  
The PGMTTO3X Command

PGMTTO3X Example

## About This Chapter

This chapter is intended for beginning users of ETU. It explains the file basic transfer commands included with ETU. You will learn the purpose of each command, as well as how to use them.

There are three primary host objects you can transfer with ETU. They are host files (physical files, logical files, and \*SAVF files), spool files, and executable programs.

There are three primary micro-based objects you can transfer with ETU. They are data files, print files, and executable programs.

For each file type, ETU has one command that transfers a host object to a microcomputer, and another command that transfers a microcomputer object to the host. The commands described in this chapter and the functions they provide are:

- FROM3XB--Transfers host files to the micro.
- TO3XB--Transfers micro-based data files to the host.
- PRTFROM3X--Transfers host spool (print) files to the micro.
- PR TTO3X--Transfers micro-based print files to the host.
- PGMFROM3XB--Transfers compiled host programs to the micro.
- PGMTTO3XB--Transfers compiled host programs from the micro to the host.

This chapter is an introduction to the file transfer commands. Refer to Chapter 10, ETU Command Reference, for complete information about a particular command.

## Transferring Data Files

The ETU commands FROM3XB and TO3XB transfer data files between the host and a microcomputer.

### Transferring Data Files From Host to a Microcomputer:

The FROM3XB Command

With the FROM3XB command, you can transfer and translate data from an AS/400 physical, logical, or \*SAVF file to a file on the micro.

To transfer a file using FROM3XB, specify the name of the host file you want to transfer, the name of the microcomputer file to which it will be transferred, and optionally, the translation information, which includes the translation type, translation table, and file definition or record length.

If you are transferring a source physical file to the microcomputer, you can remove the 12-byte sequence and date fields at the beginning of each record by specifying \*YES as the STRIP parameter. The sequence and date fields are not used by microcomputers.

If you prefer not to have the data translated during the FROM3XB operation, you can use the translation command XLATETO prior to using FROM3XB.

If you translate the file with the XLATETO command, you must specify TRANSLATE (\*NONE) in the FROM3XB command. See Chapter 4, Learning About ETU, for details about the translation types.

ETU supports off-line storage of micro-based data (via the TO3XB command). To transfer a file residing on the host and containing off-line micro-based data, specify \*SAVE as the value in the FILETYPE parameter of the TO3XB command (also specify \*NONE for the value in the TRANSLATE parameter).

FROM3XB can be run only from the micro to which the file is to be transferred. Figure 5-1 displays the FROM3XB prompt screen.

### FROM3XB Example

Suppose there is a physical file on the host called SALES that contains a member called USER1. The file was previously translated using the XLATETO command. The Mac file is stored in a folder called STORE1 and is called SALES. In this example, you will transfer this file to a Mac with no translation.

1. Choose option 1 from the ETU Main menu. The FROM3XB screen displays.
2. Enter the host file name at the File name prompt:

**sales**

Use the default, \*LIBL, as the library name. \*LIBL searches all libraries in your library list for the specified file.

3. Enter the member name at the Member name prompt:

**user1**

4. Specify \*NONE as the parameter at the Translation table file prompt because you have already translated the file with the XLATETO command.

5. Enter the Mac folder and file name at the Qualified file name prompt:

**store1:sales**

6. Because you did not specify a data translation, you can disregard the remaining parameters.

Once you have entered the appropriate information, press ENTER to start the command.

Figure 5-1 The FROM3XB prompt screen

```
Transfer from 3X: (FROM3XB)

Type choices, press Enter.

File name: . . . . .
  Library name: . . . . .      *LIBL      Name
Member name: . . . . .      *FILE      Name, *LIBL
*SAVE file type?: . . . . .  *NO        Name, *FILE, *FIRST, *LAST...
Strip seq/date field?: . . .  *NO        *YES, *NO
Type of translate?: . . . . . *TEXT      *YES, *NO
Translation table file name: . TRANSLATE   *NO, *TEXT, *DIF, *BASICS...
  Library name: . . . . .      *LIBL      Name, TRANSLATE, MACTAB
Member name: . . . . .      ETOA1      Name, *LIBL
Record length (0=existing): . . *FILE      Name
DDS or F & I file name: . . .  *NONE      0-9989, *FILE
  Library name: . . . . .      *LIBL      Name, *NONE, *DDS
Member name: . . . . .      *FILE      Name, *LIBL
Qualified file name: . . . . .      Name, *FILE, *FIRST, *LAST

More...

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



## Transferring Data Files From a Microcomputer to the Host:

### The TO3XB Command

With the TO3XB command, you can transfer and translate data from a microcomputer to a file on the host.

The TO3XB command also allows you to store data on the host in ASCII. This serves as an alternative means of off-site storage for your microcomputer data. To store the microcomputer data on the host in ASCII, specify \*SAVE as the value for the FILETYPE parameter. To transfer the data back to the microcomputer, use the FROM3XB command with \*YES as the value for the SAVTYPE parameter.

To transfer a file using TO3XB, specify the name of the micro file you want to transfer, the name of the host file to which it will be transferred, and optionally, the translation information that includes the translation type and translation table.

If the host file will be created during the TO3XB operation, you must specify the host file type, and either a record length or the file and member containing Data Description Specifications (DDS). If the host file existed prior to the TO3XB operation, ETU uses the existing file definition. For more information about file definitions, see Chapter 4, Learning About ETU.

If you prefer not to have the data translated during the TO3XB operation, you translate the file, after it is transferred to the host, with the XLATEFROM command. The XLATEFROM command translates a data file on the host after the file has been received from the micro with the TO3XB command. You can use XLATEFROM only if the following TO3XB parameters were coded as shown:

**filetype(\*data) srcseq(\*no) translate(\*none)**

The TO3XB command can be run only from the microcomputer on which the file being transferred resides.

Figure 5-2 The TO3XB prompt screen

```
Transfer to 3X: (TO3XB)

Type choices, press Enter.

File name: . . . . . Name
Library name: . . . . . *LIBL Name, *LIBL
Member name: . . . . . *FILE Name, *FILE
Record length (0=existing): . . 0-9989
File type: . . . . . *DATA *DATA, *SRC, *SAVE, *SAVF
Add seq/date field?: . . . . . *NO *YES, *NO
Type of translate?: . . . . . *TEXT *NO, *TEXT, *DIF, *BASICS...
Translation table file name: . . TRANSLATE Name, TRANSLATE, MACTAB
Library name: . . . . . *LIBL Name, *LIBL
Member name: . . . . . ATOE1 Name
Truncate text: . . . . . *NO *YES, *NO
DDS or F & I file name: . . . . *NONE Name, *NONE, *DDS
Library name: . . . . . *LIBL Name, *LIBL
Member name: . . . . . *FILE Name, *FILE, *FIRST, *LAST
Qualified file name: . . . . .

More...

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

## TO3XB Example

In this example, you will transfer a file on your PC to a new host physical file. The file on the PC is called `c:\sales\january\week1.sal`. The host physical file is called JANSAL1 in a member called REG1. To define the host file, use the existing host file definition (DDS) contained in member SALE in the source physical file QDDSSRC. Use the translation type \*BASICS.

1. Choose option 4 from the ETU Main menu. The TO3XB screen displays.
2. Enter the host file name at the File name prompt:

**jansal**

3. Enter the host member name at the Member name prompt:

**reg1**

4. Use the default, \*DATA, for the File type because the host file being created with this command is a physical file.

5. Use the translation type that translates BASIC Sequential files. At the Type of translate prompt, enter:

**\*basics**

6. Use the translation table member that translates fixed-length records from ASCII to EBCDIC. Use the default for Translation table file name. For the translation table member, enter:

**atoe1**

7. Enter the name of the source file and member that contain the file definition for the new file.

At the FFD or F & I prompt, enter:

**qddssrc**

At the member prompt, enter:

**sale**

8. Enter the microcomputer file name at the Qualified file name prompt:

**c:\sales\january\report1.doc**

9. Once you have entered the appropriate information, press ENTER to start the command. ETU will transfer the data to the host and translate it as you specified.

## Transferring Print Files

The ETU commands PRTFROM3X and PRTTO3X allow you to transfer print files between the host and the microcomputer. Once the file is transferred to the target system, you can print it on the target system's printer.

**Mac users:** The commands PRTFROM3X and PRTTO3X work only with PC-DOS and should not be used with the Mac.

### Transferring a Host Spool File to a PC Print File:

The PRTFROM3X command allows you to transfer a host spool file to a PC so it can be printed on the PC's printer. PRTFROM3X performs three functions. First, it reformats a spool file into a physical file. Next, it translates the physical file into ASCII so the file can be used on the PC. Finally, it displays a transfer prompt screen, on which you specify the name of the PC file to which you want the data transferred.

### About Host Spool Files

Host spool files reside on output queues, waiting to be released to a printer. They are stored in a format that is recognizable only to a host printer. Before you can transfer a spool file to a PC, it must be changed into a physical file with the AS/400 CPYSPLF command. You can run CPYSPLF independent from, or as part of, the PRTFROM3X command.

To transfer a spool file to the PC, enter the name of the output queue on which the spool file resides. When the output queue displays, specify which spool file you want to transfer by entering either 1 or 2 next to the file.

Entering 1 tells ETU to perform the AS/400 command CPYSPLF and translate the file for the PC. You will then be prompted for a name of a file on the AS/400 in which to temporarily store the data until you transfer it to the PC (using the FROM3XB command).

Entering 2 next to the spool file tells ETU to perform the AS/400 command CPYSPLF, translate the file for the PC, and prompt you for the PC file in which to transfer the data.

**NOTE:** Always use the translation table member ETOA1 in file TRANSLATE with this command. This table transfers spool files from EBCDIC to ASCII.

## PRTFROM3X Example

In this example, there is a host spool file on the host output queue USER1. You will transfer the file to the PC so you can print it on the PC printer. Translate the file to a PC using translation table member ETOA1 in the TRANSLATE file.

1. Choose option 3 from the ETU Main menu. The PRTFROM3X screen displays (Figure 5-3).

2. Enter the output queue name at the Output queue prompt:

**user1**

Use \*LIBL for the Output queue library name.

3. Use the defaults for the translation table file and member names, TRANSLATE and ETOA1, respectively.

Once you have entered the appropriate information, press ENTER. The USER1 output queue displays (Figure 5-4).

Figure 5-3 The PRTFROM3X prompt screen

Host spool file to micro: (PRTFROM3X)		
Type choices, press Enter.		
Output queue: . . . . .		Name
Library name: . . . . .	*LIBL	Name, *LIBL
Translation table file name: . .	TRANSLATE	Name, TRANSLATE, MACTAB
Library name: . . . . .	*LIBL	Name, *LIBL
Member name: . . . . .	ETOA1	Name
Bottom		
F3=Exit	F4=Prompt	F5=Refresh
F24=More keys	F12=Cancel	F13=How to use this display

4. Choose the spool file you want to transfer.

Press the TAB key until the cursor is next to the spool file you want to transfer. In the left-most column, type a 2 (Copyprt and transfer) next to the print item.

Figure 5-4 The PRTFROM3X Print Queue Screen

1/03/03 14:15:31		Select "COPYPRT" File				NLRISCLIFF			
Output queue: PRT01		Library: *LIBL							
Type options, press Enter.		1=Copyprt		2=Copyprt & Transfer					
Opt	File	User	User Data	Sts	Pages	Copies	Form	Type	Pty
	QPJOBLOG	QPGMR	QXOTJOB	RDY	1	1	*STD		5
	QPJOBLOG	QSYS	QPASVRP	RDY	1	1	*STD		5
	QPJOBLOG	QPGMR	QSYSWRKJOB	RDY	1	1	*STD		5
	QPJOBLOG	QPGMR	QFSIOPJOB	RDY	1	1	*STD		5
	QPSRVDMP	QTCP	QTLPD11518	RDY	1	1	*STD		5
	QPJOBLOG	QTCP	QTLPD11518	RDY	1	1	*STD		5
	QPJOBLOG	QTCP	QTPST17265	RDY	1	1	*STD		5
	QPJOBLOG	QTCP	QTLPD11518	RDY	1	1	*STD		5
	QPJOBLOG	QTCP	QTLPD11518	RDY	1	1	*STD		5
	QPJOBLOG	QPGMR	QSTRUPJD	RDY	6	1	*STD		5
	QPJOBLOG	QSYS	QWCBTCLNUP	RDY	1	1	*STD		5
	QPJOBLOG	QSYS	QWCBTCLNUP	RDY	1	1	*STD		5
	QPJOBLOG	QPGMR	QDBZMULTI	RDY	1	1	*STD		5 +
F3=Exit		F5=Redisplay							

5. Press ENTER to start the CPYSPLF procedure.

When the CPYSPLF procedure is complete, the Transfer prompt screen displays (Figure 5-5). The name of the new host physical file (PCPRINT) and the library in which it resides (QTEMP) are displayed for your convenience.

Figure 5-5 The PRTFROM3X Transfer prompt screen

Transfer from 3X: (FROM3XB)	
Type choices, press Enter.	
File name: . . . . .	> PCPRINT      Name
Library name: . . . . .	> QTEMP      Name, *LIBL
Type of translate?: . . . . .	> *NONE      *NO, *TEXT, *DIF, *BASICS...
Qualified file name: . . . . .	> 'C:\PCPRINT.LST'
Bottom	
F3=Exit	F4=Prompt
F5=Refresh	F12=Cancel
F24=More keys	F13=How to use this display

6. The name of the PC file to which the data will be transferred is also displayed. You can change the file name or leave it at the default. The default is C:\PCPRINT.LST.

Press ENTER to begin the transfer operation.

## Transferring a PC Print File to the Host:

The PRTT03X command allows you to transfer PC-DOS print files to the host and add them to a host output queue.

For this command, supply the PC file name, the host output queue name, and optionally, host printer information such as lines per inch, characters per inch, and number of copies.

Once you have supplied the information, ETU will transfer the file to the host and add the file to the AS/400 output queue that you specify. The file will print when it reaches the top of the output queue, unless you put a hold on it by entering \*YES in the HOLD parameter.

**NOTE:** Always use the translation table member ATOE3 in file TRANSLATE with this command. This table transfers print files from ASCII to EBCDIC.

Figure 5-6 The PRTT03X prompt screen

```
Spool file from micro to host: (PRTT03X)

Type choices, press Enter.

Qualified file name: . . . . .

Output queue: . . . . .          Name
  Library name: . . . . .      *LIBL   Name, *LIBL
Form length--lines/page: . . . . 66     1-255
Lines per inch: . . . . .      6       6, 8, 4, 9
Characters per inch: . . . . .  10     10, 12, 15
Form width: . . . . .          132    1-198
Overflow line number: . . . . . 63     1-255
Form type: . . . . .           *STD   Character value, *STD
Number of copies: . . . . .    1       1-99
Hold spool file: . . . . .    *NO    *YES, *NO
Translation table file name: . .  TRANSLATE  Name
  Library name: . . . . .      *LIBL   Name, *LIBL
Translation table member name:   ATOE3    Name

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

## **PRTTO3X Example**

In this example, you will transfer a PC print file to the host and print it on the host printer. The PC file is called a:\report.prt. The host output queue to which you will send the file is called USER1. Use the defaults for all other parameters.

1. Choose option 6 from the ETU Main menu. The PRTTO3X screen displays.
2. Enter the name of the PC file at the Qualified file name prompt:

**a:\report.prt**

3. Enter the output queue name at the Output queue prompt:

**user1**

4. Use the defaults for all other parameters.
5. Once you have entered the appropriate information, press ENTER.

The PC print file will be transferred to the host and placed on the output queue USER1.

## Transferring Host Programs

The ETU commands PGMFROM3XB and PGMT03XB allow you to transfer compiled executable host programs between a microcomputer and the host.

You can use these commands to transfer host programs to a micro so you can either run the host program on the micro, if you have appropriate PC software, or transfer the program to another host system. In the latter case, the PC serves as a link between two host systems.

To use these commands, you must sign on as the security officer, or be authorized to use the AS/400 programs QSCMATPG and QSCCRTPG.

**NOTE:** The PGMFROM3XB and PGMT03XB commands are no longer compatible with versions of ETU prior to version 5.02. If you've transferred a program to the PC using a previous version of ETU, you cannot transfer it back to the AS/400 using ETU 5.02 or later.

**NOTE:** Do not use these commands to transfer host source code. To transfer source code, use the FROM3XB and TO3XB commands.

### Transfer Host Programs to a Microcomputer:

#### The PGMFROM3XB Command

With the PGMFROM3XB command, you can transfer an executable host program to the micro.

To transfer a host program to a microcomputer, you supply the name of the microcomputer file in which to transfer the program, and the name of the host program and library.

This command does not translate the host program file from EBCDIC to ASCII. Because there is no translation, the file will be stored on the PC in EBCDIC.

Figure 5-7 The PGMFROM3X prompt screen

```
Transfer 3X pgm to micro file: (PGMFROM3XB)

Type choices, press Enter.

Qualified file name: . . . . .

Program name: . . . . .          Name, generic*
Library name: . . . . .          *LIBL      Name, *LIBL
Display status messages?: . . . *YES       *YES, *NO

F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

Bottom
```



If you are using this command to transfer a program between two different AS/400s having differing versions of OS/400, you must provide the Target Release Level (TGRLS) from the SAVOBJ command. To access the target release, press F10 when prompting the PGMFROM3XB command.

### **PGMFROM3XB Example**

Suppose you have a program on the AS/400 that you want to transfer to the micro. The host program is called BETA1 and is in the USER library. You will transfer it to a PC file called a:\beta1.pgm.

1. Choose option 2 from the ETU Main menu. The PGMFROM3XB screen displays.

2. Enter the name of the PC file at the Qualified file name prompt:

**a:\beta1.pgm**

3. Enter the host program and library name at the Program name prompt.

For the program, enter:

**beta1**

For the library, enter:

**user**

4. Once you have entered the appropriate information, press ENTER and the program will transfer to the PC.

## Transfer Host Programs to the Host:

The PGMT03XB command transfers executable host programs from a file on the micro to the host. This command transfers only those executable host programs from the micro that were originally transferred to the micro with the PGMFROM3XB command.

To transfer a host program from a microcomputer, supply the name of the microcomputer file that contains the program, and the name of the host program and library to receive the file.

If the program name (USERPGM) already exists in the library you specify, you will be prompted with this information and asked to delete the existing program or cancel the transfer request.

No translation is necessary when you use this command. That's because the program you are transferring from the micro will already be in EBCDIC and will remain in EBCDIC on the host.

**NOTE:** The PGMT03XB command is no longer compatible with versions of ETU prior to version 5.02. If you've transferred a program to the PC using a previous version of ETU, you cannot transfer it back to the AS/400 using ETU 400 version 5.02 or later.

Figure 5-8 The PGMT03XB prompt screen

```
Transfer micro file to 3X pgm: (PGMT03XB)

Type choices, press Enter.

Qualified file name: . . . . .

Library name: . . . . . QGPL          Character value
Display status messages?: . . . *YES    *YES, *NO

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

## PGMTO3XB Example

In this example, you will transfer a program from one host system to another host system. To do this, you would normally first transfer the host program to a PC using the ETU command PGMFROM3XB. Assume, for this example, that this step has already been done.

You will now transfer the program to a second host system using the PGMTO3XB command. To do this, copy the PC file containing the host program to a floppy diskette. Insert the diskette into a PC attached by emulation to the second host, and start ETU.

The file on the PC is called a:\beta1.pgm. You will transfer it to a file on the host called USERPGM in the QGPL library.

1. Choose option 5 from the ETU Main menu. The PGMTO3XB screen displays.
2. Enter the PC file name at the Qualified file name prompt:

**a:\beta.pgm**

3. Enter the name of the host file at the Program name prompt:

**userpgm**

Use the default library name, QGPL.

4. Once you have entered the appropriate information, press ENTER and the program will transfer to the host.

This concludes this chapter on how to transfer files between the host and the PC. For more information on each command see Chapter 10, Command Reference.

# TRANSLATING FILES

## About This Chapter

### Why Use the Translation Commands

Translation Commands Can Run in Batch Mode

Translation Commands Can Be Run From a Host Terminal

Translation Commands Reduce Connection Time

### Translating Data Files

Translating a Microcomputer File to a Host Format:  
The XLATEFROM Command

XLATEFROM Example

Translating a Host File to Microcomputer Format:  
The XLATETO Command

XLATETO Example

### Translating Print Files

Translating PC Print Files:  
The PRTXFRMPC Command

PRTXFRMPC Example

Translating Host Spool Files:  
The PRTXTOPC Command

PRTXTOPC Example

## About This Chapter

This chapter explains the ETU translation commands. These commands are used in conjunction with the ETU transfer commands. If, when you transfer a file, you specify no translation, you will translate it using one of the commands described in this chapter.

There are four ETU translation command:

- XLATEFROM--Translates a microcomputer ASCII file to a host physical file format. This is for use with micro files that were previously transferred to the host using the TO3XB command with \*NONE as the translation type;
- XLATETO--Translates a host physical file to ASCII. You can then transfer the file to the micro using the FROM3XB command with \*NONE as the translation type;
- PRTXFRMPC--Translates a host print file to a PC print file format prior to transferring the file with the PRTFROMPC command;
- PRTXTOPC--Translates a PC print file to a host print file format after you transfer the file to the host with the PRTTOPC command.

These commands can be run interactively from the micro, or in batch mode from any micro or host terminal.

The advantages to using the translation commands are described on the following page. If you are new to ETU, you should read Chapter 4, Learning About ETU, prior to reading this chapter.

## Why Use the Translation Commands

In Chapter 5, you learned about the ETU transfer commands. Those commands have two purposes. First, they transfer files between the micro and the host. Second, they translate the data so that both systems, the host and the micro, can use it.

This chapter discusses ETU translation commands. These commands have only one purpose--and that is also one of the purposes of the transfer commands--to translate the data so that both systems can use it.

Since the transfer commands can transfer and translate the data, you may wonder why the translation commands are necessary. There are three advantages to using the translation commands.

### Translation Commands Can Run in Batch Mode

The file transfer commands must be run interactively. This means that while the host is processing the ETU command, your host session is tied up and can't be used. By specifying no translation when you transfer a file, you reduce the amount of time it takes for the host to process the command. You can translate the file separately in batch mode, which frees up your micro to do other tasks.

### Translation Commands Can Be Run From a Host Terminal

Another reason for using the translation commands is that they can be run from any host terminal. The file transfer commands must be run from the micro. Again, this frees up your micro for other tasks.

### Translation Commands Reduce Modem Time

If your micro communicates with the host through a modem, you can significantly reduce the amount of phone line time if you use the translation commands. You still have to transfer the file over the modem, but by translating the file on the host prior to sending it to the micro, or after receiving it from the micro, you can save both time and money.

## Translating Data Files

The translation commands XLATEFROM and XLATETO translate files separately from the transfer commands TO3XB and FROM3XB.

These commands can be initiated from the microcomputer that contains or will contain the file, or from any host terminal. They can be run interactively or in batch mode. To run them interactively, leave the JOBID parameter blank.

There are two ways to submit the commands to batch; you can use the AS/400 command SBMJOB, or submit the command to batch within the context of the XLATEFROM and XLATETO commands by filling out the JOBID parameter.

If you use the SBMJOB command, leave the JOBID parameter blank.

### Translating a Microcomputer File to a Host Format:

#### The XLATEFROM Command

The XLATEFROM command translates a data file on the host after the file has been received from the micro with the TO3XB command. You can use XLATEFROM only if the TO3XB parameters were coded as shown:

**filetype(\*data), srcseq(\*no), translate(\*no)**

For more information on the TO3XB command see Chapter 5, Transferring Files, and Chapter 10, ETU Command Reference.

After the file is translated, it will be written to a different host file, which you specify in the TOFILE parameter. The TOFILE can be an existing file or a new file. For existing files, enter the names of the file and the member. For new files, enter the new file and member names (XLATEFROM will create these); the file type, (it can be a physical file or a source physical file); and the source file and member containing the file definition (DDS).

If you create a source physical file, you can have ETU add a 12-byte source sequence number and date field to the beginning of each record. This allows the record to be read by Source Entry Utility (SEU) after it is translated. Specify \*YES for the SRCSEQ parameter to add the sequence number and date field.

## XLATEFROM Example

Suppose you transferred a PC file to the host using the TO3XB command. You didn't translate the file during the transfer operation, but you do want to translate it now using the XLATEFROM command.

The name of the host file containing the untranslated data is RS132 in the NEWLIB library. After the file is translated, it will be written to an existing host file called HOME1, in the member of the same name. Because the host file already exists, the record length will remain at zero, and you will use the defaults for FLDDEFFILE and FLDDEFMBR. The file is in BASIC Sequential format with commas as delimiters.

1. Choose option 20 from the ETU Main menu.

The ETU Utilities menu displays.

2. Choose option 10 from the ETU Utilities menu.

The XLATEFROM screen displays.

3. At the PC data file prompt, enter the name of the file to be translated:

**rs132**

Figure 6-1 The XLATEFROM prompt screen

Translate data from PC: (XLATEFROM)				
Type choices, press Enter.				
PC data file created by FROMPC				Name
Library name: . . . . .	*LIBL			Name, *LIBL
PC data member name: . . . . .	*FILE			Character value, *FILE, *LAST
Host output data file name: . .				Name
Library name: . . . . .	*LIBL			Name, *LIBL
Host output file member name: .	*FILE			Character value, *FILE
Record length (0=existing): . .				0-9989
File type: . . . . .	*DATA			*SRC, *DATA
Add seq/date field?: . . . . .	*NO			*YES, *NO
Type of translate: . . . . .	*TEXT			*TEXT, *BASICS, *DIF, *TAB
Translation table file name: . .	TRANSLATE			Name, TRANSLATE, MACTAB
Library name: . . . . .	*LIBL			Name, *LIBL
Translation table member name:	ATOE1			Character value
Truncate text: . . . . .	*NO			*YES, *NO
Job description: . . . . .				Name
Library name: . . . . .	*LIBL			Name, *LIBL
				More...
F3=Exit	F4=Prompt	F5=Refresh	F12=Cancel	F13=How to use this display
F24=More keys				



4. At the Host output data file prompt, enter the name of the file to which data will be written once it is translated:

**home1**

5. Use the defaults for the parameters PC member and Host output file.
6. At the Type of translate prompt, enter the translation type:

**\*tab**

7. Enter the name of the translation table for ASCII-to-EBCDIC translations:

**atoe1**

Use the defaults for all the other parameters.

8. Once you have entered the appropriate information, press ENTER to start the command.

## Translating a Host File to Microcomputer Format:

The XLATETO command translates a AS/400 physical file to a format usable on the microcomputer. You will use this command prior to transferring the file to the PC with the FROM3XB command.

After you translate the data, you must enter the following FROM3XB parameters as shown:

**strip(\*no) translate(\*no)**

When the file is translated, it is written to a different host file, which you specify in the PCDATA parameter. The PCDATA file can be an existing file or a new file. For existing files, enter the names of the file and the member, for the RECL parameter, specify zero.

For new files, enter the file and member names (XLATETO will create these). If you are translating a source physical file, you can remove the first 12 bytes from each record (which contain the source sequence number and date field), by specifying \*YES as the STRIP parameter.

If you are translating an ASCII file with a fixed record length, enter \*FILE in the RECL parameter. This tells ETU to use the record length of the FRMFILE.

For more information on the FROM3XB command see Chapter 5, Transferring Files, and Chapter 10, ETU Command Reference.

Figure 6-2 The XLATETO prompt screen

```
Translate data to PC: (XLATETO)

Type choices, press Enter.

Host data file name: . . . . .      Name
  Library name: . . . . .          Name, *LIBL
Host data member name: . . . . .    *LIBL      Name, *FILE, *ALL, *FIRST...
PC data file for use by TOPC: .     Name
  Library name: . . . . .          Name, *LIBL
PCDATA member name: . . . . .      *FILE      Name, *FILE
Strip seq/date field?: . . . . .    *NO         *YES, *NO
Type of translate: . . . . .        *TEXT      *TEXT, *BASICS, *DIF, *TAB
Translation table file name: . .    TRANSLATE   Name, TRANSLATE, MACTAB
  Library name: . . . . .          *LIBL      Name, *LIBL
Translation table member name:      ETOA1         Name
Record length (0=existing): . .     *FILE      0-9989, *FILE
Job description: . . . . .          Name
  Library name: . . . . .          *LIBL      Name, *LIBL
Field translation definition: .     *DDS       Name, *DDS
  Library name: . . . . .          *LIBL      Name, *LIBL

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

## XLATETO Example

Suppose there is a file on the AS/400 that you want to transfer to your Mac. Before you transfer the file, you will translate it using the XLATETO command.

For this example, the file on the host is called LOT14. It contains one member, also called LOT14. After the data is translated, it will be written to a newly created host file called DATA3. The data will ultimately be transferred to a variable-length record on the Mac. Use the file definition of the FRMFILE parameter for the PCDATA file. Submit the XLATETO command to batch by specifying a job description.

1. Choose option 20 from the ETU Main menu.

The ETU Utilities menu displays.

2. Choose option 8 from the ETU Utilities menu.

The XLATETO screen displays.

3. At the Host data file name prompt, enter the name of the file to be translated:

**lot14**

4. At the PC data file prompt, enter the name of the file to which the translated data will be written:

**data3**

5. Use the defaults for all other parameters unless otherwise instructed.

At the Type of translate prompt, enter the variable-length translation type:

**\*text**

6. At the Translation table file prompt, enter the translation table file name for Mac translations:

**mactab**

7. At the Translation table member prompt, enter the translation table name for Mac variable-length translations:

**etoal**

8. Leave the Record length at its default of 0.

At the Job description prompt, enter the name of the job description that the job use:

**user**

9. Once you have entered the appropriate information, press ENTER to start the command.

## Translating Print Files

The translation commands PRTXFRMPC and PRTXTOPC translate print files. The command PRTXFROMPC translates a micro print file that was previously transferred to the host with no translation using the TO3XB command. The PRTXTOPC translates a host spool file prior to transferring the file to the micro with the FROM3XB command. Both translation commands can be run in batch mode on the host.

The only translation that is necessary when transferring print items between the host and a micro is the ASCII/EBCDIC translation. The PC format translations of \*TAB, \*DIF, \*TEXT and \*BASICS are not used to translate print files.

**Mac users:** Do not run the PRTXFRMPC and PRTXTOPC commands. They are intended for use only with PC-DOS.

### Translating PC Print Files:

#### The PRTXFRMPC Command

The PRTXFRMPC command translates PC-DOS print files that were transferred to the host using the TO3XB command. After the file is translated, it is added to the AS/400 output queue. The file automatically prints when it reaches the top of the queue, unless you enter \*YES in the HOLD parameter. In that case, the file remains in the output queue until you release it manually.

To run the PRTXFRMPC command, supply the name of the file to translate, the output queue to send the print file to after translation, and, optionally, printer information such as lines per inch, characters per inch, and number of copies.

Always use the translation table ATOE3, which translates print data from the micro format (ASCII) to the host format (EBCDIC).

Figure 6-3 The PRTXFRMPC prompt screen

```
Translate PC file to spool: (PRTXFRMPC)

Type choices, press Enter.

PC data file created by FROMPC
  Library name: . . . . . *LIBL      Name
  PC data member name: . . . . . *FILE  Name, *LIBL
  Output queue: . . . . .              Character value, *FILE
  Library name: . . . . . *LIBL      Name
  Length--lines per page: . . . . . 66  1-255
  Lines per inch: . . . . . 6         6, 8, 4, 9
  Characters per inch: . . . . . 10    10, 12, 15
  Form width: . . . . . 132          1-198
  Overflow line number: . . . . . 63   1-255
  Form type: . . . . . *STD          Character value, *STD
  Number of copies: . . . . . 1       1-99
  Hold spool file: . . . . . *NO      *YES, *NO
  Translation table file name: . . . . . TRANSLATE  Name
  Library name: . . . . . *LIBL      Name, *LIBL
  Translation table member name: . . . . . ATOE3     Name

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

#### PRTXFRMPC Example

Suppose you previously transferred a PC print file to a host file using the TO3XB command. Now you want to print the file on the host printer. Before you can print the file, you must translate it using the PRTXFRMPC command.

For this example, the PC file residing on the host is called PRINT1. When the translation is complete, you want the file sent to the USERX output queue. Use the defaults for all other parameters associated with this command.

1. Choose option 20 from the ETU Main menu.

The ETU Utilities menu displays.

2. Choose option 11 from the ETU Utilities menu.

The PRTXFRMPC screen displays.

3. At the PC data file prompt, enter the file name:

**print1**

4. At the Output queue prompt, enter the output queue name:

**userx**

5. Use the defaults for the other parameters.

6. Once you have entered the appropriate information, press ENTER to start the command.

The translation job will be sent to a job queue. After the file is translated, the print file will be sent to the USERX output queue and will automatically print when it reaches the top of the queue.

## Translating Host Spool Files:

### The PRTXTOPC Command

The PRTXTOPC command translates host spool files that reside on an AS/400 print queue, prior to transferring them to a PC. After the file is translated, use the FROM3XB command (with no translation) to transfer the print file to the PC.

Before you transfer the file to the PC, you must change the file from a spool file to a physical file. To do this, use the AS/400 command CPYSPLF. You can run CPYSPLF independent of the PRTXTOPC command, or within the PRTXTOPC command.

If you run CPYSPLF independently, enter \*TRANSLATE in the ACTION parameter on the PRTXTOPC prompt screen. To run CPYSPLF within the PRTXTOPC command, enter \*COPYPRT in the ACTION parameter.

Figure 6-4 The PRTXTPC prompt screen

```
Copyprt spool file - PC/XLT: (PRTXTOPC)

Type choices, press Enter.

PC data file for use by TOPC: .
  Library name: . . . . . *LIBL      Name
  PCDATA member name: . . . . . *FILE  Name, *LIBL
  Action to perform: . . . . .          Name, *FILE
  Spool file name: . . . . . QPRINT    *COPYPRT, *TRANSLATE
  Spool file number: . . . . . *LAST   Name
  Job name or * for current job: *      0000-9999, *ONLY, *LAST
  User name: . . . . .                Name, *
  Job number: . . . . .                Name, ' '
  Name of "COPYSPLF" on host: . .      000000-999999, ' '
  Library name: . . . . . *LIBL        Name
  COPYFILE member name: . . . . . *FILE  Name, *LIBL
  Control character: . . . . . *FCFC    Name, *FILE
  PAGE SIZE Length--lines/page: .     *FCFC, *PRTCTL, *NONE
  "PAGE WINDOW"...FROM column: . .    001-255
  TO column: . . . . . 132             001-999
                                          More...

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

## **PRTXTOPC Example**

Suppose you have a host spool file on the host that you want to print on the PC. Before you transfer the file to the PC, you will translate it on the host to keep the transfer time to a minimum. After the spool file has been translated on the host, you will transfer it to the PC using the FROM3XB command.

The host spool file is called LETTER2. You will run the CPYSPLF command within the context of the PRTXTOPC command.

1. Choose option 20 from the ETU Main menu.

The ETU Utilities menu displays.

2. Choose option 9 from the ETU Utilities menu.

The PRTXTOPC screen displays.

3. At the PC data prompt, enter the name of the file to which the host spool file will be translated:

**print2**

4. At the Action to perform prompt, enter the type of translation:

**copyprt**

5. At the Spool file name prompt, enter the name of the spool file to be translated. The file name is the same as the device name used to create the spool file:

**letter2**

6. Leave the Spool file number at the default of

**\*LAST.**

7. Leave the Job name parameter at its default of **\***.

8. Use the default values for all other parameters.

9. Once you have entered the appropriate information, press ENTER to start the command. When the job is complete, the file will be transferred to the PC.

For more information on the commands discussed in this chapter, see Chapter 10, ETU Command Reference.

# WORKING WITH PC FUNCTIONS

## About This Chapter

Rename a Micro File:  
The RENAMEB Command

RENAMEB Example

Allocate a Micro File:  
The ALOCATB Command

ALOCATB Example

Delete an Existing File  
The DELETEDB Command

DELETEDB Example

Test For the Existence of Micro File:  
The TESTB Command

TESTB Example



## About This Chapter

This chapter explains the ETU commands that allow you to perform PC functions, such as creating and deleting PC files, from emulation. These commands let you work with microcomputer files while you are running emulation. They eliminate the need for you to return to the PC or Mac environment to create or delete a file.

Four commands are discussed in this chapter. The commands and their functions are listed for you here:

- `RENAMEB`--Renames an existing microcomputer file;
- `ALOCATB`--Allocates (creates) a new microcomputer file;
- `DELETEDB`--Deletes an existing microcomputer file;
- `TESTB`--Tests for the existence of a microcomputer file.

## Rename a Micro File:

### The RENAMEB Command

With the RENAMEB command, you can rename an existing micro file while you are in emulation mode. You supply the current name of the micro file and the new name. The new file uses the same path as the file it replaces.

### RENAMEB Example

In this example, you are working in emulation on the AS/400 and you want to rename a PC file called c:\data.pgm to c:\data1.pgm.

1. Choose option 20 from the ETU Main menu.

The ETU Utilities menu displays.

2. Choose option 2 from the ETU Utilities menu.

The RENAMEB prompt screen displays.

3. Enter the micro file name at the Qualified file name prompt:

**c:\data.pgm**

4. Enter the file's new name at the New micro file name prompt:

**data1.pgm**

5. Start the command by pressing ENTER.

ETU will rename the file on the PC.

### Figure 7-1 The RENAMEB prompt screen

```
Rename a micro file: (RENAMEB)

Type choices, press Enter.

Qualified file name: . . . . .
New micro file name: . . . . .

Display status messages ?: . . . *YES      *YES, *NO

Bottom

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

## Allocate a Micro File: The ALOCATB Command

The ALOCATB command creates a new file on the microcomputer.

### ALOCATB Example

In this example, you are in emulation and want to create a file on the PC's c: drive called sales.feb in the reports directory (c:\reports\sales.feb).

1. Choose option 20 from the ETU Main menu.

The ETU Utilities menu displays.

2. Choose option 1 from the ETU Utilities menu.

The ALOCATB prompt screen displays.

3. At the Qualified file name prompt, enter the name for the micro file you want to create:

**c:\reports\sales.feb**

4. Press ENTER to start the command.

ETU will now create the file on the PC.

Figure 7-2 The ALOCATB prompt screen

```
Allocate a micro file: (ALOCATB)
Type choices, press Enter.
Qualified file name: . . . . .
Display status messages ?: . . . *YES      *YES, *NO

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

## Delete an Existing File:

### The DELETEDB Command

Just as you may need to allocate (create) files on a microcomputer while in emulation, you may also need to delete files. The DELETEDB command deletes any micro file.

ETU will send you a completion message if it doesn't find the file you want to delete. Enter \*NO at the NOTIFY prompt if you don't want ETU to notify you.

**NOTE:** When you use the DELETEDB command, all data in the designated micro file will be permanently deleted.

Figure 7-3 The DELETEDB prompt screen

```
Delete a micro file: (DELETEDB)

Type choices, press Enter.

Qualified file name: . . . . .

Notify operator: . . . . . *YES      *YES, *NO
Display status messages?: . . . *YES      *YES, *NO

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

## **DELETEDB Example**

In this example, you are running emulation and you want to delete a file on the PC's a: drive called fiscal.doc in the 'reports' subdirectory (a:\reports\fiscal.doc).

1. Choose option 20 from the ETU Main menu.

The ETU Utilities menu displays.

2. Choose option 3 from the ETU Utilities menu.

The DELETEDB prompt screen displays.

3. At the Qualified file name prompt, enter the name of the micro file you want to delete:

**a:\report\fiscal.doc**

4. Leave the Notify operator parameter at its default of \*YES.

You will be notified whether or not the file is found.

5. Press ENTER to start the DELETEDB command.

ETU will delete the file on the PC and notify you if the file is not found by sending a message to your display.

## Test for the Existence of a Micro File:

### The TESTB Command

With the TESTB command, you can check to see if a file exists on the micro while you are in emulation. The host you sends a completion message about the existence of the file.

### TESTB Example

In this example, you are running emulation on your micro and you want to check to see if a certain file exists on the micro. The micro file is called c:\datafile.doc.

1. Choose option 20 from the ETU Main menu.

The ETU Utilities menu displays.

2. Choose option 4 from the ETU Utilities menu.

3. At the Qualified file name prompt, enter the name of the file:

**c:\datafile.doc**

4. Press ENTER to start the TESTB command.

A message will be sent to the display when the operation is complete.

Figure 7-4 TESTB prompt screen

```
Test a micro file: (TESTB)

Type choices, press Enter.

Qualified file name: . . . . .

Notify operator: . . . . . *YES          *YES, *NO, *LDA
LDA position for file size: . . . . . 1-1012
Display status messages?: . . . *YES      *YES, *NO

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

# MODIFYING TRANSLATION TABLES

About This Chapter

About Translation Tables

Modifying a Translation Table

Make a Copy of the Original Table

Editing the Translation Table

Translation Table Record Format

Sample Translation Table.

Compiling the New Translation Table Member

Naming the New Member

## About This Chapter

ETU provides you with eight translation tables which it uses to translate data. Each table performs a separate function. Some translate data from the micro to the host and others from the host to the micro. (See Chapter 4, *Learning About ETU*, for more information about the translation tables.)

The tables provided with ETU will serve the needs of most users. If the translation tables supplied with ETU do not fit your needs, you can modify an existing table. This chapter explains how to modify a translation table and how to use the modified table to translate data.

This chapter is intended only for advanced users of ETU and the AS/400. You should be familiar with using AS/400 Source Entry Utility (SEU) and compiling source on the AS/400.

To modify the translation tables, use the ETU commands STRSEU and EDITTABLE. These commands are explained in this chapter and in Chapter 10, *ETU Command Reference*.

Appendix F, *ASCII-EBCDIC Printable Characters*, lists the ASCII and EBCDIC characters and their hexadecimal equivalents. Refer to Appendix F when modifying a translation table.



## About Translation Tables

The translation tables provided with ETU have three purposes. They:

- translate data from ASCII to EBCDIC or from EBCDIC to ASCII;
- specify whether the file being translated has a fixed or variable format;
- tell the host whether the micro involved in the translation is a DOS-based PC or a Mac.

Each translation table is stored as a member in a physical file. The DOS-based tables are in a physical file called TRANSLATE in the host library containing the ETU programs. The Mac translation tables are in a physical file called MACTAB in the same library.

The source for each table is also included with ETU. Each table is stored as a source member in a source file called TRANSSRC, also in the library containing the ETU programs. To modify a translation table, you will change one of the source members.

There are four tables for PC-DOS translations and three for the Mac translations. Each table and its function is described below in Table 8-1.

Table 8-1 The ETU translation table members in source file TRANSSRC

PC DOS Translation Tables	ETOA1	The EBCDIC-to-ASCII character translation member producing a variable length record, with carriage return and line feed added after the last byte of data in each record. Trailing blanks are stripped.
	ETOA1	The EBCDIC-to-ASCII character translation member producing a basic random file on the PC, with no carriage return line feed.
	ETOA1	The EBCDIC-to-ASCII character translation member producing a fixed length record, with carriage return and line feed added at the end of each record. Trailing blanks are not stripped.
	ATOE1	The ASCII-to-EBCDIC fixed and variable-length member
	ATOE3	The ASCII-to-EBCDIC print translation member
Mac Translation Tables	ETOA1M	The EBCDIC-to-ASCII character translation member producing a variable length record, with carriage return and line feed added after the last byte of data in each record. Trailing blanks are stripped.
	ETOA2M	The EBCDIC-to-ASCII character translation member producing a variable-length record, with carriage return and line feed added after the last byte of data in each record. Trailing blanks are stripped.
	ATOE1M	The EBCDIC-to-ASCII character translation member producing a basic random file on the PC, with no carriage return line feed.

## Modifying a Translation Table

If you decide that the standard translation tables included with ETU won't work for you, you can create a new one. To create a new table, modify the source member closest in function to one of the existing members. For example, if you want to create a table that translates data from a PC to the host, modify the source member ATOE1. The source members for the translation tables are in the source physical file TRANSSRC.

Changing a translation table is a three-part process.

- First, make a copy of the source member containing the translation table you want to modify. This step keeps the original tables intact.
- Next, edit the translation table. In this step you will change the translation table using SEU so that it meets your needs.
- The first two steps in this process are performed using the command STRSEU (option 20 on the ETU Utilities menu).
- Finally, compile the new member. Using the ETU command EDITTABLE, you will compile the new translation table from a source member to a physical file member. EDITTABLE is option 21 on the ETU Utilities menu.

Once you compile the new translation table, you can use it to translate files. Each step in this process is described on the following pages.

## Make a Copy of the Original Table

Before you modify a translation table member, it is recommended that you copy the original member, which resides in the supplied TRANSSRC file, into another member. This leaves the original members inviolate in case you still need to use them.

To copy a member, use the AS/400 command STRSEU (option 20 on the ETU Utilities menu). Using SEU, copy the member you will modify into another member.

## Editing the Translation Table

Once you have copied the original source member, you can modify or edit the member using SEU, the same command used to copy the original. To edit the member, specify the name of the file containing the member to be modified, the name of the member, the type of source to be edited, and, optionally, text identifying the source member. When the source member displays, you can edit the source member using any of the normal SEU commands. For information on using SEU, see the IBM manual AS/400 SEU User's Guide/Reference, SC09-1172.

After you have made the appropriate changes to the source member, save the changes, exit SEU, and compile the source member.

In order to change the file you must be familiar with the record format within the source file. The format of the file containing the translation table members is described below.

## Translation Table Record Format

Each record in the translation table is used to translate a character in one set, ASCII or EBCDIC, to a character in the other set. For each character record you modify, you provide the following information:

- Scan character
- Replace character
- Action to be taken before or after the scan/replace
- Pad characters for EBCDIC-to-ASCII translations
- End-of-file character in ASCII-to-EBCDIC translations
- End-of-record character in ASCII-to-EBCDIC translations

The record format for the translation table source file is shown in Table 8-2.

Table 8-2 Translate Table Source Record Format

Position in the Record	Functions (Record Fields)
1-18	FROM HEX (or special function)
19-36	TO HEX
37	Translate before or after performing control
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-to-EBCDIC translation
44-80	Comment
81-96	Not used

The fields in the record are described in detail below:

**FROM HEX** From one to nine characters expressed in hexadecimal representation. The translation program scans the input data for the FROM HEX character. When found, it replaces the data with 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.

**NOTE:** Only \*TEXT reformatting supports the translation of the full nine-byte (nine-character) string in the table. \*DIF, \*BASICS, and \*TAB reformatting programs supports only a single-byte (one-character) translation string in the table.

The FROM HEX table size limits are listed in Table 8-3.

Table 8-3 FROM HEX Table Size Limits

Number of FROM Characters	Maximum Number of Table Entries			
	*TEXT	*DIF	*BASICS	*TAB
1	256	256	256	256
2	50	xxx	xxx	xxx
3	50	xxx	xxx	xxx
4	40	xxx	xxx	xxx
5	30	xxx	xxx	xxx
6	20	xxx	xxx	xxx
7	15	xxx	xxx	xxx

For example, the table above indicates that a translation table can contain a maximum of 20 "From" entries with a string length of 6 bytes. The xxx markers indicate that the length is not supported for that type of translation. Normally, all translation types will use only the single character translation.

**TO HEX** From zero to nine 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**

Determines when the translation is performed relative to performing control action. (This field is labeled XBA on the compiler report.) 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 a TO3XB 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**

Marks the end of the current record being translated upon detection of the FROM HEX characters (used in FROMPC or TO3XB translation). It can be either Y or a value encoded \*1 through \*9.

"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              C1C1C1C2          Then write AAAB
```

### **End-of-File Action**

This is identical to end-of-record action with one exception. This function also completes translation and terminates the program.

### **Comments**

Are printed in the translation table compiler listing.

There are five special records for the translation table compiler. These records define what action is to be taken for end-of-record and end-of-file for FROM3XB translation. On TO3XB translation, the special records determine if trailing blanks or nulls should be omitted.

The \*1 to \*9 records specify additional TO HEX characters for 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 FROM3XB 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 upon end of file on the file being translated. This can be used to write the end-of-file character hex 1A in the record before closing the file to be sent to the micro with the FROM3XB command. The \*EOF is coded in position 1 through 4 of the translation table source.

**\*TRAIL statement** -- This statement specifies the trailing character to be stripped in a FROM3XB translate, and the default pad character for a TO3XB 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 one through nine 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 translation table source by coding \*\* in positions 1 and 2.

## Sample Translation Table

The following is an example of an ASCII-to-EBCDIC translation table. For a complete listing, see member ATOE1 in the file TRANSSRC in the library containing ETU. It can be printed with the Source Entry Utility.

Figure 8-4 Sample ASCII-to-EBCDIC Translation Table

FROM HEX	TO HEX	XBA	PAD	EOR	EOF	Comments
01						Omit unprintable characters
02						
.						
.						
30	F0					Numeric Characters
31	F1					
32	F2					
33	F3					
34	F4					
.						
.						
41	C1					Alphabetic characters
42	C2					
43	C3					
.						
.						
0DOA		B				CR/LF; End-of-record
1A		B				End of file

## Compiling the New Translation Member

After you create or modify a new translation source file member using SEU, you must compile it into a physical file member before you can use it in the ETU translation program. The compiler

- edits the translate table source;
- prints a listing of the table (including any error messages);
- writes the compiled table to a physical file member.

To compile the new member, use the AS/400 command EDITTABLE (option 20 on the ETU Utilities menu). Supply the name of the source file and member to be compiled. Then supply the name of the physical file and member into which to compile the new member.

## Naming the New Member

The decision of what to name the new physical file and member is an important one. This is because the ETU programs that use translation tables have defaults for the translation table file and member names. The default for the translation file is always TRANSLATE. The default for the member name depends on the ETU command being used. If you fail to take into account the default names when you name the new physical file and member, you will most likely have to change the defaults each time you translate a file with the newly created table.

While having to change the file and member name presents no real problems, you can save time when defining ETU transfers by following these suggestions.

We suggest that you store any new PC-based translation table members in the file TRANSLATE.

If you modify an existing table that will replace the original table, you can give the new table the same name as the existing table. By doing this, the ETU commands that previously used the original table as the default will use the new table as the default.

If you don't intend for the new table to replace the original table, then be sure to give the new table a unique name. Then, each time you translate a file using the new translation table, be sure you specify the new table name in the TRANSMBR prompt.



# ADVANCED ETU FEATURES

About This Chapter

Multiple-device Transfer

Transferring From Multiple Micros

Using PWRT03XB

Transferring a Host File to Multiple Micros

Micro-to-Micro Transfers

The PCFROMPC Command

The PCTOPC Command

Transferring Files Between Multiple Macs

Mac File Transfer Examples

If InterAccess is located at the root level of the hard drive

If InterAccess is one or more folders deep

Record and File Selection

About Creating Auxiliary Storage Pools

Transferring Selected Fields

Field Selection - Example 1

Field Selection - Example 2

Transferring Selected Records

Record Selection - Example 1

Record Selection - Example 2

Entry and Exit Programs - Example

Delete Files After Transfer

Saving Request Definitions

Re-using Request Definitions

Performing Batch Transfers

Batch Transfer - Example

Time and Date Scheduling - Example

## About This Chapter

Most ETU commands allow you to transfer a file interactively between the host and a micro. This chapter describes features that allow advanced users to go beyond the basic ETU operations.

The features included in this chapter are part of the ETU power commands: PWRFROM3XB, PWRTO3XB, PCFROMPC, and PCTOPC.

The commands PWRFROM3XB and PWRTO3XB provide the same functionality as their more basic ETU counterparts, FROM3XB and TO3XB. In addition, the power commands provide the following features:

- Multiple-device support -- Allows you to simultaneously transfer a file to or from multiple micros with one command.
- Record and field selection -- Transfers only the records and fields you specify from the host to the micro.
- Entry and exit program support -- Executes user-defined programs within the context of the power command.
- File deletion -- Deletes a file on the source system after that file is transferred to the target system.
- Savable request definitions -- Allows you to save and reuse the requests you make with these commands.
- Batch transfers -- Allows you to submit a command to a job queue for processing in batch mode.
- Time and date scheduling -- Allows you to submit a job to batch and schedule a time in the future when the job will be released for processing.

The PCFROMPC and PCTOPC commands allow you to transfer files from one micro to another micro.

This chapter explains these functions and gives examples of situations in which you might use them. For more information on the transferring and translating files with these commands, see the FROM3XB and TO3XB commands in Chapter 5.

The commands associated with the features in this chapter are described in detail in Chapter 10, Command Reference.

## Multiple-device Transfer

Multiple-device file transfer is a feature supported by all ETU power commands. Using multiple-device transfer, you can:

- Transfer a single host file to multiple micros (PWRFROM3XB);
- Transfer a file from multiple micros to the host (PWRT03XB);
- Transfer a single micro file to multiple micros (PCTOPC);
- Transfer a file from multiple micros to a single micro (PCFROMPC).

(The command you use to perform the described operation is shown in parentheses.)

**NOTE:** When transferring to or from multiple micros, all micros involved in the transfer must be of the same platform, either Mac or DOS-based PC.

As an example of when you would use multiple-device transfer, suppose you have five PCs, each attached to the host through emulation. Using multiple-device transfer, you can transfer a file from each PC to the host with one command.

All multiple-device transfers can be run interactively or in batch mode. See Batch Transfers later in this chapter for more information on running these commands in batch mode.

**NOTE:** To transfer a file to or from multiple devices, you must be authorized to use each device specified in the transfer. In addition, each device must be powered on, have emulation loaded, and be at the AS/400 sign-on screen.

### Transferring From Multiple Micros

With the PWRT03XB command, you can transfer a file from one or more micros to the host. To transfer from more than one micro, specify the name of the micro file to be transferred, the device name of the micro, and the name of the host file to transfer it to. The parameters used with PWRT03XB multiple device transfer are described below.

**Qualified file name** The name of the micro file. In order to transfer a file from multiple micros, the file must be identically named (including the path) on each micro.

**MICLST** The device name of the source micro. To transfer from more than one micro, enter the name of each device. (You must be authorized to use each device you specify in the MICLIST parameter. If you can sign on to the device, then you are authorized to use it.)

To enter more than one device name, enter a plus sign (+) here and press ENTER. This displays multiple MICLST prompts. The default, \*REQUESTER, transfers a file from the micro to which you are

signed on.

Each device you transfer from must be powered on, have emulation loaded, and be at the AS/400 sign-on screen.

**NOTE:** This parameter must be changed from \*REQUESTER when you run the PWRT03XB command in batch mode.

When you transfer the files to the host, you can specify how you want the file stored:

- In one physical file with multiple members;
- In one physical file with one member (data is stored together);
- In multiple physical files with one member each.
- To store the files in one physical file with multiple members, specify the following:

**TOFILE** Enter a file name.

**Member name** \*MULTI  
The member names will be the same as the device names from which the file was transferred.

To store the files in one physical file with a single member, specify the following parameters as shown:

**TOFILE** Enter a file name.

**Member name** \*FILE  
The name of the file and member will be the same as you specify in the TOFILE file parameter, or you can specify a separate member name.

To store the files in multiple physical files with a single member in each:

**TOFILE** \*MULTI

**Member name** \*FILE  
The file and member names are the same as the device names from which the file was transferred. For example, if you transfer a file from a micro with the device name PRT2AD3, the file and member are both named PRT2AD3.

You must specify Data Description Specifications when you request \*MULTI in the TOFILE parameter.

## Using PWRT03XB

In order to send files from multiple micros to the same host library using PWRT03XB with the TOFILE(\*MULTI) option, you must rename any files sent in a first transfer before performing a second transfer.

This procedure is necessary because when files are transferred from multiple micros to a single micro, the resulting file names on the destination micro are the device names from which the data was transferred. For example, a file sent from Device 1 will end up on the receiving micro as a file named device1. A file sent from Device 2 will be nameddevice2.

Because of this naming convention, any files sent in a second file transfer (from the same-named micros to the same host library) will end up with exactly the same names as the files sent in the first transfer (device 1, device 2, etc.). As a result, the files sent in the second transfer will overwrite the data sent in the first transfer.

In addition, with PWRT03XB you may also use the option to delete the micro file after transfer to the host, in which case, after a second transfer the data from the first transfer will no longer exist on either the micro or the host.

There are two ways to avoid copying over files in the second transfer: You must either use the host function to rename the first files sent to the host, or specify the option ADDRPL (\*ADD). Note that ADDRPL (\*ADD) combines data from multiple micro files in a single data file on the host.

## Transferring a Host File to Multiple Micros

The PWRFROM3XB command allows you to transfer a single host file to up to ten micros with one command. To transfer a file to multiple micros, specify the name of the host file to be transferred, the name of the micro file to which the data will be transferred, and the device names of the micros to which to transfer the data. The parameters used with PWRFROM3XB multiple-device transfers are described below.

- FRMFILE** The name of the host file that you want to transfer.
- Qualified File Name** The name of the file on the micro to which to transfer the file. Any path name you enter must already exist on the micro (ETU does not create subdirectories).
- MICLIST** The device name of the target micro. The default, \*REQUESTER, transfers the file to the micro from which you initiate the command.

To transfer the file to multiple micros, enter the name of each device. (You must be authorized to use each device you specify in the MICLIST parameter. If you can sign on to the device, then you are authorized to use it.) Enter a plus sign (+) and press ENTER to display multiple MICLIST prompts.

Each target device must be powered on, have emulation loaded, and have the AS/400 sign-on screen displayed.

**NOTE:** This parameter must be changed from \*REQUESTER when you run the PWRFROM3XB command in batch mode.

## Micro-to-micro Transfers

In addition to allowing file transfers between a micro and a host, ETU allows you to transfer files directly from one micro to another micro. Use the ETU commands PCFROMPC and PCTOPC to transfer files between multiple micros.

The PCFROMPC command allows you to transfer a file from multiple micros to a single micro. The PCTOPC command allows you to transfer a file from one micro to multiple micros.

### The PCFROMPC Command

To transfer a file from multiple micros to a single micro, specify the name of the file to be transferred, the device or devices from which the file is being transferred, and the device to which the file is to be transferred.

As with PWRT03XB above, in order to send files from multiple micros to the same destination directory using PCFROMPC, you must rename any files sent in a first transfer before performing a second transfer.

This procedure is necessary because when files are transferred from multiple micros to a single micro, the resulting file names on the destination micro are the device names from which the data was transferred. For example, a file sent from Device 1 will end up on the receiving micro as a file named device1. A file sent that was from Device 2 would be nameddevice2.

Because of this naming convention, any files sent in a second file transfer (from the same-named micros to the same receiving-micro directory) will end up with exactly the same names as the files sent in the first transfer (device 1, device 2, etc.). As a result, the files sent in the second transfer will overwrite the data sent in the first transfer. To avoid this, use PWRRENAMEB to rename the files resulting from the first transfer.

The parameters used with PCFROMPC multiple-device transfer are described below:

**Qualified file name** The name of the file to be transferred. If you are transferring a file from multiple micros, the file name, including the path name, must be identical on each micro.

**MICLST1** The device name of the source micro(s). To transfer a file from more than one micro, enter the device name of each. (You must be authorized to use each device you specify in the MICLST1 parameter. If you can sign on to the device, then you are authorized to use it.)

Enter a plus sign (+) here and press ENTER to display multiple MICLST1 prompts in which to enter the device names.

Each device you transfer from must be powered on, have emulation loaded, and have the AS/400 sign-on screen displayed.

**MICLST2** The device name of the target micro. The default, \*REQUESTER, transfers multiple files to the micro from which you initiate the command. (You must be authorized to any device you specify in this parameter or the transfer will fail.)

Files transferred to the target micro have the same names as the devices from which they came. Device names can be up to ten characters long. If the device name of the source micro is longer than eight positions, the file name on the target micro will be in the format: XXXXXXXX.XX.

For example, a file transferred from a micro with device name Port5 Addr3 to another micro is called PORT5ADD.R3.

<p><b>NOTE:</b> MICLST2 must be changed from *REQUESTER to the target micro name when you run this command in batch mode.</p>
---



## The PCTOPC Command

To transfer a single file from one micro to multiple micros, specify the name of the file, the device name of the micro from which the file is being transferred, and the name of the device or devices to which the file is being transferred.

**QFNAME1** The name of the micro file to be transferred. If you are transferring a file to multiple micros, the file name, including the path name, must be identical on each micro. For additional information, see About Microcomputer Files in Chapter 3.

**MICLST1** The device name of the source micro. The default, \*REQUESTER, transfers the file from the micro on which the PCTOPC command is currently being run. (To transfer a file from a different micro, enter the device name of the micro. You must be authorized to use the device you specify in this parameter, or the transfer will fail.)

<p><b>NOTE:</b> This parameter must be changed from *REQUESTER to the name of the source micro when you run this command in batch mode.</p>
---

**MICLST2** The device name of the micro to which the file is being transferred. You can transfer the file to up to ten micros. (To transfer the file to more than one micro, enter the device name of each micro. You must be authorized to use each device you specify in this parameter, or the transfer will fail.)

Enter a plus sign (+) here and press ENTER to display multiple MICLST2 prompts.

Each device you transfer to must be powered on, have emulation loaded, and have the AS/400 sign-on screen displayed.

## Transferring Files Between Multiple Maces

If you are transferring files between your Mac and the host, or you are transferring files between Maces that share the same hard drive name, you may use the file transfer example on page 3-11. However, if you are transferring files between Maces that have unique hard drive names, you must use either of the examples below.

Before attempting to transfer files using multiple Maces, verify that the following elements are identical on each involved Mac:

- the location of InterAxxcess (NLynx Technologies' Mac emulator), whether it's located on the hard drive or several folders deep;
- the path through which the file is to travel.

Once you have verified that you are maintaining each of these conditions, you are ready to transfer files using the examples below as guidelines.

### Mac File Transfer Examples

In the two examples below, assume the following:

- you are using the PCTOPC command (described on page 10-14) to transfer a file named Test from your Mac to two other Maces;
- Test resides in a folder named Training, which is located on the root level of your hard drive;
- each of the receiving Maces have unique hard drive names, and a Training folder already exists on each of their root directories.

#### If InterAxxcess is located at the root level of the hard drive

In this example, InterAxxcess is on the root level of each Mac's hard drive (Figure 9-1).

1. From the ETU menu, issue the PCTOPC command.
2. Enter the parameters as you normally would, taking special care with the qualified pathname, as described in step 3.
3. For the qualified pathname, enter:

**:Training:Test**

The colon before Training represents the folder in which InterAxxcess resides. Note that there is only one colon, since InterAxxcess resides at the root level of the hard drive.

Once the transfer is complete, each of the receiving Maces will have a new file named Test in their Training folders.

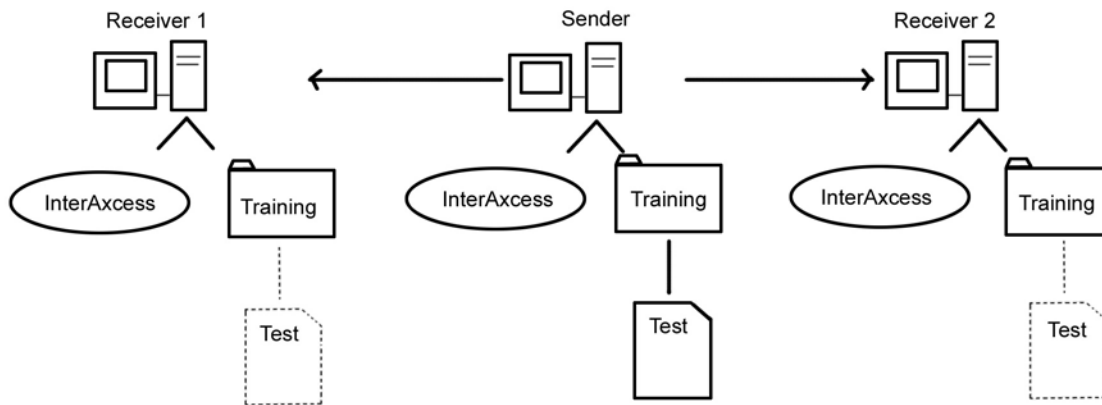


Figure 9-1 Transferring Files With InterAccess on the Root Level of the Hard Drive

### If InterAccess is one or more folders deep

In this example, on each Mac, InterAccess is located in a folder named AS/400, which is in turn located in a folder named Apps. The Apps folder resides at the root level of the hard drive (Figure 9-2).

1. From the ETU menu, issue the PCTOPC command.
2. Enter the parameters as you normally would, taking special care with the qualified pathname, as described in step 3.
3. For the qualified pathname, enter:

**:::Training:Test**

The colons before Training represent the following: The first colon represents the AS/400 folder, the second represents the Apps folder, the third represents the root level of the hard drive. Note that the single colon between training and test represents that the file Test resides in the folder Training, and again, the colon before Training represents the hard drive.

Once the transfer is complete, the receiving Macs will each have a new file named Test in their Training folders.

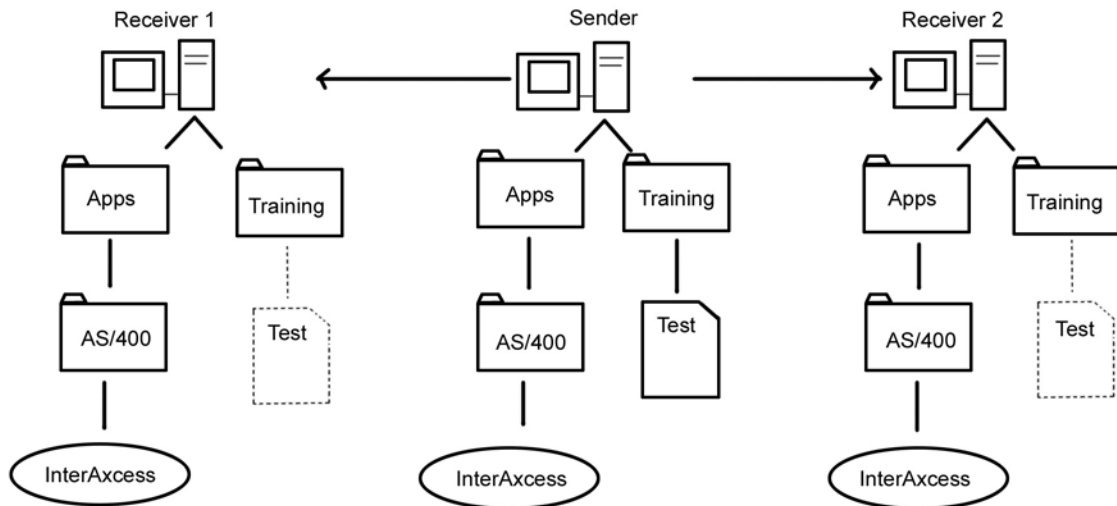


Figure 9-2 Transferring Files With InterAccess Several Levels Deep

## Record and Field Selection

Record selection and field selection are features that allow you to transfer only the records and fields you specify. This feature is part of the PWRFROM3XB command. Record selection and field selection are not mutually exclusive functions of ETU, although it's not necessary to use them together.

**NOTE:** Record and field selection can be used only with externally-defined host physical files.

## About Creating Auxiliary Storage Pools

The Record and Field Selection feature within the PWRFROM3XB command will not work properly if you define one or more additional Auxiliary Storage Pools on the AS/400. This is because ETU creates temporary work files in the library QTEMP, which is always located in Auxiliary Storage Pool #1 (which comes with OS/400). If you create an additional Auxiliary Storage Pool, ETU cannot create the work files associated with the data file that is stored in any newly created Auxiliary Storage Pools.

If you do create a new Auxiliary Storage Pool, you can simulate record and field selection in a file transfer operation by creating a logical file or an open query file and transferring those files.

## Transferring Selected Fields

When you transfer a physical file from the host to a micro with the PWRFROM3XB command, you have the choice of transferring all the fields in the file, or, using field selection, transferring only the fields you select. You can also specify the order in which the data will reside in the micro file.

For each field you select to transfer, supply information to the following parameters, found under the heading FLDSEL (field selection) on the PWRFROM3XB prompt screen. The parameters are:

**Field name** The name of the field you want to transfer. You may transfer up to 25 fields.

**Key field** Specifies whether the field is to be a key field. The default, \*NONE, does not re-sort the records, but rather, leaves them in sequential order. The field does not have to be an existing key field in the file.

The total of the key fields cannot exceed 120 characters in length.

**Ascending or descending** Specifies whether the records are sorted in ascending or descending order (\*ASC or \*DES).

## Field Selection -- Example 1

Suppose there is a file on your host computer that contains information about the employees in your company. Each record in the file contains a variety of information about one employee, such as the employee's name, address, phone number, rate of pay, and length of service.

In this example, you will transfer only the names and phone numbers of each employee to a file on the PC. You also want the PC file to list the employees in alphabetical order, from A to Z. Table below illustrates how the data will appear once it has been transferred to the PC.

Table 9-1 Transfer data for Example 1.

<i>Employee name</i>	<i>Phone number</i>
Anderson	449-6771
Brandybuck	449-4971
Dilbert	449-3228
Haram	434-3434
Martinez	234-3556
Simpson	323-0497
Spock	449-2323
Stiles	323-0165
Wally	232-3232

To transfer only the employee name and phone number fields in the order shown above, follow these steps:

1. Enter the file transfer and translation information as you would with any file transfer.
2. Enter a plus sign (+) next to the FLDSEL parameter and press ENTER. This displays multiple prompts.
3. In the first prompt, for field name, enter EMPNAM(employee name). For the key field, enter \*KEY1. This tells ETU that the employee name field is a key field with the highest sort priority (and the only sort priority, in this example). For ascending or descending, enter \*ASC. This tells ETU to sort the records from A to Z. (If you entered \*DES, the records would be sorted in order from Z to A.)
4. In the second prompt, for field name, enter PHONE (phone number). For the key field, use the default, \*NONE, because the file will not be sorted using the PHONE field. The Ascending or
5. Descending field is not used because the Key Field value is \*NONE.

When you execute this command, the file will transfer to the PC. The only fields that transfer are the employee name and phone number. The information will be sorted into alphabetical order.

## Field Selection -- Example 2

In this example, you will see how you can use more than one key field when transferring selected fields. You will transfer the same file as in the first example.

This time, you will transfer the employee names and their rates of pay. You will sort the information first in order of highest salary to lowest salary, and second, in alphabetical order when two or more employees earn the same salary. Table 9-2 below illustrates how the file will appear on the PC.

Table 9-2 Transferred data for Example 2

<i>Salary</i>	<i>Employee name</i>
10,000	Spock
15,000	Anderson
20,000	Stiles
30,000	Brandybuck
65,000	Dilbert
75,000	Wally
80,000	Martinez
90,000	Haram
100,000	Simpson

1. Enter the file transfer and translation information as you would with any file transfer.
2. In the FLDSEL parameter, next to the field name prompt, enter a plus sign (+) and press ENTER. This displays multiple prompts.
3. In the first prompt, for field name, enter PAYRAT (rate of annual pay). For key field, enter \*KEY1. This tells ETU that the PAYRAT field is a key field with the highest sort priority. For ascending or descending, enter \*DES. This tells ETU to sort the records in the order of the highest salary to the lowest salary.
4. In the second prompt, for field name, enter EMPNAM. For key field, enter \*KEY2. This sorts any employees who earn the same salary in alphabetical order. Enter \*ASC to sort the employees from A to Z.

## Transferring Selected Records

ETU allows you to transfer selected records from a host file to a file on a micro. You select the records to be transferred by creating test cases. A test case is a set of criteria you specify within the PWRFROM3XB command. You can specify up to 15 test cases. If a record meets the criteria of a test case, it is transferred.

For instance, if you are transferring the employee file described earlier, you can create a test case that transfers only the records of those employees making more than \$20,000. You could describe this test case as follows: If the value in the PAYRAT field is greater than 20,000, then transfer the record.

The PWRFROM3XB command contains four parameters under the heading RECSEL (record selection). For each test case you create, you will supply the relationship, the field to be tested, the relational operator, and the comparison value. You can create up to fifteen test cases for a single file transfer. These parameters are explained below.

- Relationship** Relates one test case to another. The first test case must use \*IF. The default, \*NONE, transfers all records in the file.
- The other relationships are \*AND, \*OR, \*ANDIF, \*ORIF and are used in situations where more than one test case is evaluated.
- Field to be tested** Specifies the name of the field to be tested. Use any field name from the file that was specified in the FRMFILE parameter. If you use field selection, you must specify a field to be tested.
- Relational operator** Specifies the relationship between the field to be tested and the comparison value. The operators are:
- \*EQ Equal to
  - \*NE Not equal to
  - \*GT Greater than
  - \*LT Less than
  - \*LE Less than or equal to
  - \*GE Greater than or equal to
- Comparison value** Specifies the value to which to compare the data in the field to be tested. Enter a numeric value for numeric fields, and an alphabetic value for alphabetic fields. Alphabetic values must be enclosed in single quotation marks (').

**NOTE:** If you specify an alphabetic comparison value in which a space is used to separate words or characters (e.g., 'red apple'), the records will transfer as expected. However, due to a limitation in the AS/400 operating system, a saved request definition with a comparison value as described will not work properly.



## Record Selection -- Example 1

Suppose you want to transfer the employee file (described in the previous section) from the host to the PC. You will transfer only the records of those employees earning more than \$20,000 per year. To do this you will create a test case that transfers the records that meet the following criterion: the value in the PAYRAT field is greater than 20,000.

1. Enter the file transfer and translation information as you would for any file transfer.
2. In the RECSEL Relationship parameter, enter **\*IF**.
3. Enter **PAYRAT** for the field to be tested. Enter **\*GT** (greater than) as the relational operator. Enter 20000 for the comparison value.

When you initiate the command, the records that meet the above criteria are transferred.

## Record Selection -- Example 2

This example shows you how you can use more than one test case to select records. It uses the same employee file as in the other examples.

Suppose you want to transfer a file from the host to a PC. You will transfer only the records that meet the following set of criteria: the value in the PAYRAT field is greater than 20000 and the employee has five or more years of service with the company.

1. Enter the file transfer and translation information as you would with any file transfer.
2. Enter a plus sign (+) next to the RECSEL parameter and press ENTER. This displays 15 prompts, each containing the following parameters: relationship, field to be tested, relational operator, and comparison value.
3. In the first prompt, for the relationship, enter **\*IF**. Enter **PAYRAT** for the field to be tested. Enter **\*GT** (greater than) as the relational operator. Enter **20000** for the comparison value.
4. In the second prompt, enter **\*AND** as the relationship. Enter **YEARS** for the field to be tested. Enter **\*GE** (greater than or equal to) as the relational operator. For the comparison value, enter **5**.

When you initiate the command, all records meeting both sets of criteria will be transferred to the PC.

## Entry and Exit Programs

With the PWRFROM3XB and PWRT03XB commands, you can run any user-defined programs before and/or after the file is transferred. Programs that run before the file transfer occurs are called entry programs. Programs that run after the file is transferred are called exit programs.

To use the entry and exit programs, enter the name of the program and the library in which it resides in the ENTRYPGM and EXITPGM parameters.

**NOTE:** Entry and exit programs do not have to be ETU programs. Any program you can run on the AS/400 can be executed as an entry or exit program. Remember that if you run a power command in batch mode, any entry and exit programs must also be able to run in batch.

### Example

Suppose you have a payroll file on the host that you want to transfer to a PC. Before you transfer the file, you want to update it with current payroll information by running a program that outputs to the payroll file.

At the end of the file transfer operation, you want to run a program that sends a message notifying the system operator that the file transfer is complete.

Enter the file transfer and translation information as you would for any file transfer, using the PWRFROM3XB command. In the ENTRYPGM parameter, enter the name of the entry program to be run, along with the library in which it resides. Do the same for the EXITPGM parameter.

When you execute the PWRFROM3XB command, the following events occur in the order shown:

- the entry program runs and updates the payroll file;
- the file is transferred to the PC;
- the exit program runs sending a message to the system operator.

## Deleting Files After Transfer

The commands PWRFROM3XB and PWRT03XB allow users to delete files after the files have been transferred.

Both commands contain a parameter called DLTFILE. If you enter \*YES in the DLTFILE parameter in the PWRFROM3XB command, the host file you are transferring to the micro is deleted after the transfer occurs.

If you do not have the authorization to delete files on the host, or if there are logical files attached, ETU will not allow you to delete them even if you enter \*YES in the DLTFILE parameter.

With the PWRT03XB command, the file on the micro is deleted after it is transferred to the host. To do this, enter \*YES as the value for the DLTFILE parameter.

## Saving Request Definitions

Some ETU commands allow you to save the definition of the current request. The request definition is the completed, or filled-out, list of parameters for the command you are running. You can re-use the saved definition later to run the same command.

The ETU commands that allow you to save request definitions are PCFROMPC, PCTOPC, PWRFROM3XB, PWRT03XB, FROM3XB, and TO3XB.

These commands all contain a parameter called SAVRQS. To save the definition for the current request, enter \*YES for this parameter and enter the request name you desire. The request name can be up to ten characters long.

As an example, suppose you regularly transfer a weekly sales file from a PC to the host, using the PWRT03XB command. By saving the definition of that request, you can avoid re-keying the same information each time that you transfer the file.

## Re-using Request Definitions

Once you have saved a request definition, you can re-use it at any time, with the ETU command VWSAVRQS (view saved requests). Using this command, you can run the saved requests interactively or submit them to batch (only the ETU power commands). You can also change or delete any of the saved requests. You may work with only those saved requests that you created. Only the security office can view requests from another user.

The ETU command VWSAVRQS displays all the requests that you have saved. To view the saved requests, enter VWSAVRQS on the AS/400 command line. This displays only the requests saved by the user running the VWSAVRQS command. To take action on one or more requests, enter one of the following options next to the requests.

Figure 9-3 VWSAVRQS prompt screen

```
1/06/03          View saved request definitions

Options:  1= Execute      2= Submit to batch  3= Change
          4= Delete      5= Extended view

Rqs name   Cmd name   Cmd parms
PWRT03XB   QFNAME1('c:\bootlog.txt') MICLST(NLRISCLIF...

F1= Unfold/Fold      F3= Exit
```

The options and the actions they represent are:

- 1 **Execute** Interactively runs the command using the saved definition.
- 2 **Submit to batch** Prompts for the job description parameters and submits the request to batch. Batch commands cannot be longer than 256 characters. In order to run a job in batch mode, the device you specify must be powered on, running emulation, and displaying the AS/400 sign-on screen.

<p><b>NOTE:</b> Only the ETU power commands can be submitted to batch; not the commands FROM3XB or TO3XB.</p>
---

- 3 **Change** Allows you to change the saved request.
- 4 **Delete** Deletes the specified request.
- 5 **Extended view** Displays an extended view of the request. Only 45 characters can be displayed on the screen. If the saved request is longer than 45 characters, it will be truncated to 45 characters and followed by an ellipsis (...). Option 5 shows the entire request definition.

**Fold/Unfold** The function key F1 provides an alternative view of the highlighted request. It shows the name of the user who created the definition, the device from which it was created, the creation date, and creation time. It also lists the name of the user who last changed the definition, the workstation from which it was changed, and the date and time that the change occurred.

## Performing Batch Transfers

The ETU transfer commands featured in this chapter (PWRFROM3XB, PWRT03XB, PCFROMPC, PCTOPC, ) can all be run in batch mode from either a microcomputer emulating a 5250 workstation or a dumb terminal.

To submit one of these commands to batch, use the ETU command SBMBATXFER. Enter SBMBATXFER on the AS/400 command line, press F4, and provide the information described below.

Figure 9-3 The Submit Batch Transfer prompt screen

```
Submit batch transfer: (SBMBATXFER)

Type choices, press Enter.

Command to submit to batch: . . . . . Name
Job name: . . . . . *JOBD Name, *JOBD
Job description: . . . . . *USRPRF Name, *USRPRF
Library name: . . . . . *LIBL Name, *LIBL
Job queue: . . . . . QBATCH Name
Library name: . . . . . *LIBL Name, *LIBL
Hold submitted job ?: . . . . . *NO *NO, *YES

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

**SBMCMDNAM** The name of the ETU command you want to submit to batch. Enter PCFROMPC, PCTOPC, PWRFROM3XB, or PWRT03XB.

**JOBNAME** The name of the job you are submitting to batch. The default, \*JOBID, assigns the name of the job description from the user's AS/400 user profile. You can enter any name up to ten characters.

**JOBID** The job description. The default, \*USRPRF, uses the job description associated with the user's AS/400 profile.

**Library name** The name of the library containing the job description. The default, \*LIBL, searches all libraries in the user's library list.

**JOBQ** The name of the AS/400 job queue to which the command will be sent. The default, QBATCH, is an IBM-supplied job queue.

**Library name** The name of the library containing the job queue. The default, \*LIBL, searches all libraries in the user's library list.

**HOLD** Specifies whether the job will be held or not once it is placed in the job queue. The default is \*NO. To hold the job, enter \*YES. You must then manually release the job.

After you supply the above information and press ENTER, you are prompted with the command name you specified in the SBMCMDNAM parameter. Fill out the parameters for the command as you would if you were running the job interactively.

If you are submitting one of the ETU power commands to batch, you must change the default value, \*REQUESTER, to the actual device address you will use.

In order to run the job in batch mode, the micro or micros you are transferring to or from must be powered on, running emulation, and displaying the AS/400 sign-on screen.

## Batch Transfer – Example

In this example, you will submit an ETU command to batch mode while you are working from a dumb host terminal. You will transfer a file from one PC to another PC using the PCFROMPC command. You will submit the command to run in batch mode using the SBMBATXFER command.

The file you will transfer is called a:\test2.doc and is on a PC whose device name is PRT3AD3. You will transfer the file to another PC whose device name is PRT3AD1. In order for the transfer to be successful, both devices must be powered on, running emulation, and at the AS/400 sign-on screen.

1. Enter the SBMBATXFER command on the AS/400 command line. The SBMBATXFER prompt screen displays.
2. At the SBMCMDNAM prompt, enter the name of the ETU command you want to submit to batch:

**pcfrompc**

3. Use the default values for the parameters JOBNAME, JOBID, and JOBQ.
4. Press ENTER. The PCFROMPC prompt screen displays.
5. Enter the name of the micro file to be transferred:

**a:\test2.doc**

6. Enter the name of the device from which the file will be transferred:

**prt3ad3**

7. Enter the name of the device to which the file will be transferred:

**prt3ad1**

8. Press ENTER to execute the command.

The PCFROMPC command will be submitted to the default job queue, QBATCH.

## Time and Date Scheduling

The ETU Time and Date Scheduling utility allows you to submit a job to batch and specify a time in the future that the job will be released to the job queue. The utility can be used to schedule jobs when you know the host system is less busy than normal, such as on weekends or at night.

All ETU power commands and any other AS/400 or user-defined command or program that can be run in batch mode can be scheduled using the Time and Date Scheduling feature.

To use Time and Date Scheduling, enter the ETU command SBMSCHJOB (submit scheduled job) on the command line, and press F4. The Submit Scheduled Job prompt screen displays (Figure 9-4).

Figure 9-4

Submit scheduled job: (SBMSCHJOB)

Type choices, press Enter.

Command: . . . . .		
Hour in 24 hour clock format: .		00-24, *IMMED
Minute (00-59): . . . . .	00	00-59
Date in system format: . . . . .	*TODAY	000000-999999, *TODAY, +1...
Job name: . . . . .	*JOBID	Name, *JOBID
Job description: . . . . .	QBATC	Name
Library name: . . . . .	*LIBL	Name, *LIBL
Bottom		
F3=Exit	F4=Prompt	F5=Refresh
F24=More keys	F12=Cancel	F13=How to use this display

At the Command prompt enter the command name that you want to schedule followed by any valid parameters, or prompt for the command's parameters by entering a question mark (?) followed by the command name.

Then enter the time and date at which you want the job to be released, the job name, and the job description. If you entered the entire command string, the job will be submitted to the job queue and held until the scheduled time.

If you prompted for the parameters, the command prompt screen will display. After you define the request for the command, it is placed in the job queue on hold until the scheduled time, when it is released for processing.

**NOTE:** In order for the scheduling function to work, the job scheduling program JOBSCHED, must be active on the AS/40 subsystem QINTER. See the SBMSCHJOB command in Chapter 10 for information on starting JOBSCHED.



## Time and Date Scheduling – Example

Suppose you want to transfer a file to a PC from the host using the PWRFROM3XB command. You want this command to run later the same night in batch mode. The job name is NIGHT and the job description is QBATCH.

1. On the command line, enter the ETU command that schedules the job.

**sbmschjob**

2. Press **F4**.

The Submit Scheduled Job prompt screen displays.

3. Prompt for the command you want to schedule:

**pwrfrom3xb**

4. Enter the time that the job will be released to the job queue: for the hour:

**22**

for the minute:

**30** (This will release the job at 10:30 p.m.)

5. Since you want the job to run on the same day you are submitting it, leave the date at its default of \*TODAY.

6. Enter the job name and job description:

for the job name:

**night**

for the job description:

**qbatch**

Press ENTER after you enter the appropriate information, and the PWRFROM3XB prompt screen displays.

7. Enter the transfer information in the PWRFROM3XB prompt and press ENTER.

The job is submitted to the job queue and will be released at the specified time.

## *Part Three*

# Command Reference

"Command Reference" is an alphabetical presentation of all AS/400 native mode ETU commands and the parameters and values associated with each command.

## **10-1          Command Reference**

# ETU COMMAND REFERENCE

ALLOCATB Command

DELETEDB Command

EDITTABLE Command

EDTUDF Command - EDTUDF Notes

FROM3XB Command - FROM3XB Notes

PCFROMPC Command

PCTOPC Command

PGMFROM3XB Command

PGMTO3XB Command

PRTFRO3XB Command

PRTTO3XB Command

PRTXFROMPC Command

PRTXTOPC Command

PWRDELETEDB Command

PWRFROM3XB Command

PWRRENAMEB Command

PWRTO3XB Command

RENAMEB Command

SBMBATXFER Command

SBMSCHJOB Command - To Schedule a Job

STRSEU Command

TESTB Command

TO3XB Command

VWSAVRQS Command - VWSAVRQS Notes

XLATEFROM Command

XLATETO Command

## ALOCATB Command

<b>Parameter description</b>	<b>Parameter name</b>	<b>Parameter values</b>
Qualified file name	QFNAME	micro file name
Display status message?	DSPSTMSG	*YES or *NO

The ALOCATB command allocates (creates) an empty file on the micro.

**QFNAME** The name of the micro file to be allocated. For additional information, see About Microcomputer Files in Chapter 3.

**DSPSTMSG** Displays program status messages during the execution of the command. The default is \*YES. To turn the messages off, enter \*NO.

## DELETEB Command

Parameter description	Parameter name	Parameter values
Qualified file name	QFNAME	micro file name
Notify Operator?	NOTIFY	*YES or *NO
Display status message?	DSPSTMSG	*YES or *NO

The DELETEB command deletes a file on the micro.

**QFNAME** The name of the micro file to be deleted. For additional information, See About Microcomputer Files in Chapter 3.

**NOTIFY** Specifies whether you will be notified if the file to be deleted is not found. This is done by sending a message to your message queue.

**\*YES** You will be notified if the file is not found (default).

**\*NO** You will not be notified if the file is not found.

**DSPSTMSG** Displays program status messages during the execution of the command.

**\*YES** Status messages are displayed (the default).

**\*NO** Status message are not displayed.

## EDITTABLE Command

Parameter description	Parameter name	Parameter values
Source member	SRCMBR	ETOA2 ETOA3 ATOE1  ATOE3 ETOA1M ETOA2M ATOE1M translation table member name
Source file	SRCFILE	TRANSSRC source file name Library name *LIBL library name
Translation table file name	TRANSFILE	TRANSLATE MACTAB file name Library name *LIBL library name
Translation table member name	TRANSMBR	*SRCMBR member name

The EDITTABLE command compiles an ETU translation table after you have edited it with the STRSEU command. A listing of the source table is printed, indicating any errors detected. The physical file member can then be named as the TRANSMBR parameter in any of the commands involving file translation.

**SRCMBR** The name of the source member containing the translation table. The following translate table members are provided with ETU.

### PC-DOS Tables

**ETOA1** EBCDIC-to-ASCII variable length records

**ETOA2** EBCDIC-to-ASCII fixed length records

**ETOA3** EBCDIC-to-ASCII suitable for government and banking applications

**ATOE1** ASCII-to-EBCDIC variable & fixed length records

**ATOE3** ASCII-to-EBCDIC print records

### Mac Tables

**ETOA1M** EBCDIC-to-ASCII variable length records

**ETOA2M** EBCDIC-to-ASCII fixed length records

**ATOE1M** ASCII-to-EBCDIC

**SRCFILE** The qualified name of the source physical file containing the translation table source. The default value is TRANSSRC.

**Library** The name of the library containing the SRCFILE \*LIBL is a valid library.

**TRANSFILE** The qualified file name of the physical file to contain the compiled table. It is created if not found. The default file is TRANSLATE in the library containing the ETU programs.

**Mac Users:** If you use the TRANSFILE default (TRANSLATE), the member names will have an **M** on the end of the name (for example, ATOE1M). The Mac tables are also shipped in a translate file called MACTAB. The members in this file do not have the **M** at the end of the name (for example, ATOE1).

**Library** The name of the library containing the TRANSFILE. \*LIBL is a valid library.

**TRANSMBR** The member name for the compiled table. If the default of \*SRCMBR is specified, the name used will be the same as the one specified for the SRCMBR parameter.

## EDTUDF Command

The EDTUDF command allows you to add user-defined options to the ETU Main menu (WSMENUS). Any valid AS/400 command can be added to the menu using this function. Additionally, ETU utilities, such as EDTUDF, can be added to the ETU menu.

To add a user-defined option, enter the command EDTUDF on the AS/400 command line and press F4. You will first be prompted to enter a user profile. You can add user-defined functions for a single user or a group of users. To add a user-defined function for a single user, enter the user's profile name. To add a user-defined function for a group of users, enter the group profile name. To add user-defined functions for yourself, type EDTUDF on the command line and press Enter.

When the Edit User-defined Functions screen displays, supply a description, function, and mode. After you have completed adding the user-defined option, press F3 to return to the main menu.

<p><b>NOTE:</b> In order for the new menu option to display on the ETU Main menu you must exit the ETU menu (press F3) and redisplay the ETU menu by issuing the WSMENUS command. The Edit User-defined Functions parameters are described below.</p>
---

**Description** The description of the user-defined function as it will appear on the ETU Main menu. It may be up to 32 character long.

**Function** The AS/400 command that will be run when the menu option is issued. It may be up to 512 characters long.

**Mode** Specifies how the command will be executed when selected from the ETU Main menu. Valid entries are:

**\*PMT** The user is prompted after choosing the option.

**\*EXE** The user-defined option executes immediately after the user chooses the option.

When you specify \*EXE for the mode, you are prompted to enter values for the command you specified in the Function parameter. Each time you choose the menu option, the command executes using the same values for the parameters as specified when the user-defined function was created.



## **EDTUDF Notes**

Up to ten user-defined functions can be added to the ETU menu. The first time you start the EDTUDF utility (where no user-defined functions have previously been created), the utility will be in "add" mode. While in add mode, you may create as many user-defined functions as you like, up to ten.

Each time you start the EDTUDF utility, (after you add one or more user-defined functions) it will be in "change mode". To place EDTUDF in add mode, use the F9 function key. While in change mode, you can change or delete a user-defined function, or display an extended view of an existing user-defined function.

To change a user-defined function, enter 1 next to the function you want to change and press ENTER. Change the function by typing over the existing text.

To delete a user-defined function, enter 2 next to the function you want to delete and press ENTER.

To display an extended view of a function, enter 3 next to the function and press ENTER. Extended view allows you to display function strings that are longer than 20 positions. If the function is longer than 20 positions, it will appear on the EDTUDF screen followed by an ellipsis (...).

## FROM3XB Command

Parameter description	Parameter name	Parameter values
File name	FRMFILE	host file name translation table member name
Library name		*LIBL library name
Member name	FRMMBR	*FILE *FIRST *LAST *ALL
*SAVE file type?	SAVTYP	*NO or *YES
Strip seq/date file	STRIP	*NO or *YES
Type of translation	TRANSLATE	*TEXT or *YES *NO or *NONE *DIF *BASICS *TAB
Translation file name	TRNSFILE	TRANLATE MACTAB file name
Library name		*LIBL library name
Member name	TRANSMBR	*ETOAl member name
Record length	RECL	*FILE 0-9989
DDS or F&I file name	FLDDEFFILE	*NONE or *DDS file name
Library name		*LIBL library name
Member name	FLDDEFMBR	*FILE file name
Qualified file name	QFNAME1	micro file name
Save request definition?	SAVRQS	*NO or *YES
Request name		name of request
Display status message?	DSPSTMSG	*NO or *YES

The FROM3XB command transfers data from an AS/400 physical or logical file member to a file on the micro and, optionally, translates the data according to the specified translation type and translation table.

Four different types of translation can be selected: \*DIF, \*BASICS, \*TAB, and \*TEXT (see Chapter 4, Learning About ETU, for more information.). See FROM3XB Notes below.

**FRMFILE** The qualified file name of the AS/400 file that is to be transferred to the micro.

**Library** The library containing the physical file to be transferred. The default, \*LIBL, searches all libraries in the user's library list.

**FRMMBR** The FRMFILE member name. You can enter the member name or one of the following values:

**\*FILE** Transfers the member with the same name as the FRMFILE file (the default).

**\*FIRST** Transfers the first member arrival sequence in the file.

**\*LAST** Transfers the last member arrival sequence in the file.

**\*ALL** Transfers all members from the FRMFILE file into one micro file. \*ALL is not valid with \*DIF translation.

**SAVTYP** Used to transfer files that were previously transferred to the host with the \*SAVE file type using the TO3XB command.

**\*NO** The file being transferred was not previously transferred to the host with the \*SAVE file type (the default).

**\*YES** The file being transferred was previously transferred to the host with the \*SAVE file type.

When using \*SAVE to move programs, all necessary data (including Mac data and resource forks) is transferred.

**STRIP** For source physical files only: Specifies whether the first 12 bytes of each record are to be ignored when reading the file. This 12-byte area contains the sequence number and date fields for source physical files.

**\*NO** Do not strip off any data from the record (the default).

**\*YES** Strip off (ignore) the first 12 bytes.

**TRANSLATE** Determines whether the data is to be translated, and if so, what type of translation is to be performed.

**\*TEXT** or **\*YES** The AS/400 file is to be translated into ASCII text format. The default is \*TEXT.

**\*DIF** The AS/400 file is to be translated into DIF format.

**\*BASICS** The AS/400 file is to be translated into BASIC Sequential format.

**\*TAB** The AS/400 file is to be translated into BASIC Sequential format with tab characters as the field delimiters.

**\*NO** or **\*NONE** No translation of the AS/400 file is to be performed. (Usually this is used only after an independent translation has been done using the XLATETO command.)

**TRANSFILE** The qualified file name of the translation table file to be used when a translation is specified. The default is TRANSLATE.

**Mac Users:** There are two files you can use. If you use the default file (TRANSLATE), the member names have an **M** on the end of the name (for example, ETOA1M). The Mac tables are also shipped in a file called MACTAB. The members in this file do not have the **M** at the end of the name (for example, ETOA1).

**Library** The library containing the source for the translation tables.  
\*LIBL is the default.

**TRANSMBR** The TRANSFILE member name. The default name is ETOA1, an EBCDIC-to-ASCII table supplied with ETU. ETOA2 should be used for fixed-length record translations.

**RECL** The record length of the translation work file. Valid record length entries are:

**\*FILE** The record length of the FRMFILE file is used.

**1-9989** The record length for the translation work file.

If an ASCII fixed-record-length translation is to be performed, specify \*FILE to produce the desired file. If you are not translating to ASCII fixed-length records, you may specify any record length that could allow for better performance by buffering the data.

**FLDDEFFILE** Specifies the file that defines the file being transferred. If the file being transferred was not externally defined and the translation type is either \*DIF, \*TAB, or \*BASICS, you must specify a source physical file containing either physical file data description specifications (DDS), or file and input (F&I) specifications.

If the file being transferred was previously defined with either DDS or F&I specifications, or is a \*TEXT translation type, use the default, \*NONE, or its equivalent, \*DDS.

**FLDDEFMBR** The FLDDEFFILE member name. You can enter the member name or one of the following values:

**\*FILE** Transfers the member with same name as the FLDDEFFILE.

**\*FIRST** Transfers the first member (arrival sequence) in the file.

**\*LAST** Transfers the last member (arrival sequence) in the file.

**QFNAME1** The name of the micro file to contain the data from the AS/400. For additional information, See About Microcomputer Files in Chapter 3.

**SAVRQS** Saves the current request definition so it can be used at a later time. The request definition is this completed FROM3XB command with the parameter values as you have specified them. The default is \*NO. To save the request definition, specify \*YES and enter the request name you desire.

**Request Name** The name of the SAVRQS request definition. The request name can be up to ten characters long. (To recall the saved request, use the ETU command VWSAVRQS.)

**DSPSTMSG** Displays program status messages during the execution of the command. The default is \*YES. To turn the messages off, enter \*NO.

### **FROM3XB Notes**

When you specify file translation with this command, the AS/400 file (FRMFILE) is read by the translate program. After the program translates the data, it writes the translated data into a workfile that is created for this purpose (PCTTOWORK in QTEMP). It is this workfile that will then be transferred to the micro.

If the sequence number and date fields (like those of a source physical file) exist in the qualified file, these fields can be stripped during the ASCII-to-EBCDIC translation process by specifying STRIP (\*YES).

If the data was translated prior to using the FROM3XB command, with the XLATETO or PRTXTOPC commands, you must code the FROM3XB parameters STRIP and TRANSLATE as follows:

**strip(\*no), translate(\*no)**

## PCFROMPC Command

Parameter description	Parameter name	Parameter values
Qualified file name	QFNAME	micro file name
Transfer from which micro?	MICLIST1	Device name(s) of one or more micros.
Transfer to which micro?	MICLIST2	*REQUESTER device name of target micro
Save request definition? Request name	SAVRQS	*YES or *NO Name
Display status message?	DSPSTMSG	*YES or *NO

The PCFROMPC command transfers a file from one or more microcomputers to a single microcomputer. All micros involved in a file transfer must be of the same platform, either Mac or DOS-based PC, but not both.

**NOTE:** Before using PCFROMPC to transfer multiple files to a single micro, review Transferring Files With PCFROMPC and PWRT03XB in Chapter 9 in order to minimize the risk of copying over previously transferred files.

**QFNAME1** The name of the micro file to be transferred. If you are transferring a file from multiple micros, the file name, including the path name, must be identical on each micro. For additional information, see About Microcomputer Files in Chapter 3.

**MICLST1** The device name of the source micro(s). To transfer a file from more than one micro, enter the device name of each. (You must be authorized to use each device you specify in the MICLIST1 parameter. If you can sign on to the device, then you are authorized to use it.)

Enter a plus sign (+) here and press ENTER to display multiple prompts in which to enter the device names.

Each micro from which you transfer data must be powered on, have emulation loaded, and have the AS/400 sign-on screen displayed.

**MICLST2** The device name of the target micro. The default, \*REQUESTER, transfers multiple files to the micro from which you initiate the command. (You must be authorized to any device you specify in this parameter, or the transfer will fail.)

If multiple files are transferred, the files transferred to the target micro have the same name as the device from which they came. If the host device name is longer than eight positions, the file name on the micro will be in the format: XXXXXXXX.XX.

For example, a file transferred from a micro with device name Port5 Addr3 to another micro is called PORT5ADD.R3.

**Warning:** Since ETU automatically assigns the target file names when more than one file is transferred, you may have problems with overwriting previously transferred files. To avoid this situation, change the name of the files once they are received on the target micro, or copy them to a different directory.

<p><b>NOTE:</b> This parameter must be changed from *REQUESTER to the device name of the target micro when you run the PCFROMPC command in batch mode. The device you specify must be powered on, have emulation loaded, and have the AS/400 sign-on screen displayed.</p>
--

**SAVRQS** Saves the current request definition so it can be used at a later time. The request definition is this completed PCFROMPC command with the parameter values as you have specified them. The default is \*NO. To save the request definition, specify \*YES and enter the request name you desire.

**Request Name** The name of the SAVRQS request definition. The request name can be up to ten characters long. (To recall the saved request, use the ETU command VWSAVRQS.)

**DSPSTMSG** Displays program status messages during the execution of the command. The default is \*YES. To turn the messages off, enter \*NO.

## PCTOPC Command

Parameter description	Parameter name	Parameter values
Qualified file name	QFNAME	micro file name
Transfer from which micro?	MICLIST1	Device name(s) of one or more micros.
Transfer to which micro?	MICLIST2	*REQUESTER device name of target micro
Save request definition? Request name	SAVRQS	*YES or *NO Name
Display status message?	DSPSTMSG	*YES or *NO

The PCTOPC command transfers a file from one micro to one or more other micros. All micros involved in a file transfer must be of the same platform, either Mac or DOS-based PC, but not both.

**QFNAME1** The name of the micro file to be transferred. If you are transferring a file from multiple micros, the file name, including the path name, must be identical on each micro. For additional information, see About Microcomputer Files in Chapter 3.

**MICLST1** The device name of the source micro. The default, \*REQUESTER, transfers the file from the micro to which you are currently signed on. To transfer a file from a different micro, enter the device name of that micro. (You must be authorized to use the device you specify in this parameter, or the transfer will fail.)

**NOTE:** This parameter must be changed from \*REQUESTER to the device name of the source micro when you run the PCTOPC command in batch mode. The device you specify must be powered on, have emulation loaded, and have the AS/400 sign-on screen displayed.

**MICLST2** The device name of the target micro(s). You can transfer the file to up to ten micros. To transfer the file to more than one micro, enter the device name of each micro. (You must be authorized to use the device you specify in this parameter, or the transfer will fail.) Enter a plus sign (+) here and press ENTER to display multiple prompts.

Each micro to which you transfer data must be powered on, have emulation loaded, and have the AS/400 sign-on screen displayed.

**SAVRQS** Saves the current request definition so it can be used at a later time. The request definition is this completed PCTOPC command with the parameter values as you have specified them. The default is \*NO. To save the request definition, specify \*YES and enter the request name you desire.

**Request Name** The name of the SAVRQS request definition. The request name can be up to ten characters long. (To recall the saved request, use the ETU command VWSAVRQS.)



**DSPSTMSG** Displays program status messages during the execution of the command. The default is \*YES. To turn the messages off, enter \*NO.

## PGMFROM3XB Command

Parameter description	Parameter name	Parameter values
Qualified file name	QFNAME	micro file name
Program	PGM	host program name
Library name		*LIBL library name
Display status message?	DSPSTMSG	*YES or *NO
Target release level	TGTRLS	*CURRENT see SAVOBJ command for other parameters

The PGMFROM3XB command transfers a host executable program from the AS/400 to a specified file on the micro. Do not use this command for transferring program source code. Transfer source code with the FROM3XB, PWRFROM3XB, TO3XB, or PWRTO3XB command.

**NOTE:** To use this command, you must be signed on as a security officer, or be authorized to execute the program QSCMATPG.

**QFNAME1** The name of the micro file to contain the AS/400 program. For additional information, see About Microcomputer Files, in Chapter 3.

**PGM** The qualified name of the AS/400 executable program to be transferred to the micro.

**Library** The library containing the program. The default is \*LIBL.

**DSPSTMSG** Displays program status messages during the execution of the command. The default is \*YES. To turn the messages off, enter \*NO.

**TGTRLS** The OS/400 release level of the target AS/400. To be used when transferring a program between AS/400s having differing versions of OS/400.

The syntax for entering the target release level differs from one OS/400 release level to another. You can obtain the syntax for the target release level of the AS/400 you're transferring the program to by running the SAVOBJ command. To access the SAVOBJ command, press F10 from when prompting the PGMFROM3XB command.

## PGMTO3XB Command

Parameter description	Parameter name	Parameter values
Qualified file name	QFNAME	micro file name
Program Library name	PGM	host program name *LIBL library name
Display status message?	DSPSTMSG	*YES or *NO

The PGMTO3XB command transfers a host executable program from a file residing on the micro. Do not use this command for transferring program source code. Transfer source code with the FROM3XB, PWRFROM3XB, TO3XB, or PWRTO3XB command.

**NOTE:** To use this command, you must be signed on as a security officer, or be authorized to execute the program QSCMATPG.

**QFNAME1** The name of the micro file to contain the AS/400 program. For additional information, see About Microcomputer Files, in Chapter 3.

**PGM** The qualified name of the AS/400 executable program to be transferred to the micro.

**Library** The library containing the program. The default is \*LIBL.

**DSPSTMSG** Displays program status messages during the execution of the command. The default is \*YES. To turn the messages off, enter \*NO.

## PRTFROM3X Command

Parameter description	Parameter name	Parameter values
Output Queue Library name	OUTQ	output queue name *LIBL library name
Translation table name Library name	TRANSFILE	TRANSLATE table file name *LIBL library name
Translation table member name	TRANSMBR	ETOA1 member name

The PRTFROM3X command allows you to transfer a print file from the host to a micro. This command is interactive only. It displays a list of items in a host print output queue to be reformatted for printing on the micro.

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

**OUTQ** The qualified print output queue. This option displays a print queue screen. On this screen, enter 1 or 2 next to the print item(s) to be copied for use on the micro.

**1=Copyprt** Performs the AS/400 command CPYSPLF, translates the file for the micro, and stores it in the AS/400 file which you specify on the TRANSLATE prompt screen (see Figure 10-1 on the following page). You will then transfer the file to the micro using the FROM3XB command.

**2=Copyprt & Transfer** Performs the AS/400 command CPYSPLF, translates the file for the micro, and displays a FROM3XB prompt (see Figure 10-2) for the transfer of the translated work file from the AS/400.

**NOTE:** You must specify TRANSLATE (\*NO) in the FROM3XB command. This is because the PRTFROM3X command performs the translation.

**NOTE:** If the OUTQ that you specify has a lot of files in it, the command takes more time to process.

**Library** The name of the library containing the OUTQ. \*LIBL is valid.

**TRANSFILE** The qualified file name of the translation table file to be used. The default is TRANSLATE.

**Library** The name of the library containing the TRANSFILE. \*LIBL is valid.

**TRANSMBR** The TRANSFILE member name. The default name is ETOA1, an EBCDIC-to-ASCII table supplied with ETU.

Figure 10-1 The PRTFROM3X Translate prompt screen

```
Translate spool file to PC: (TRANSLATE)

Type choices, press Enter.

Name of PCPRINT file on host: . > PCPRINT      Name
Library name: . . . . . > QTEMP              Name, *LIBL

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

Bottom
```

The screen you get with **1=Copyprt**

Figure 10-2 The PRTFROM3X Transfer prompt screen

```
Transfer from 3X: (FROM3XB)

Type choices, press Enter.

File name: . . . . . > PCPRINT      Name
Library name: . . . . . > QTEMP      Name, *LIBL
Type of translate?: . . . . . > *NONE *NO, *TEXT, *DIF, *BASICS...
Qualified file name: . . . . . > 'C:\PCPRINT.LST'

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

Bottom
```

The screen that you get with **2=Copyprt & Transfer**

## PRTTO3X Command

Parameter description	Parameter name	Parameter values
Qualified file name	QFNAME	micro file name
Output queue	OUTQ	host output queue name
Library name		*LIBL library name
Lines per page	LINES	66 lines per page
Lines per inch	LPI	6 lines per inch
Characters per inch	CPI	10 characters per inch
Form width	WIDTH	132 form width
Overflow line number	OVERFLOW	63 overflow line
Type of form label	FORMTYPE	*STD forms type
Number of copies	COPIES	1 number of copies
Place job on hold?	HOLD	*NO or *YES
Translation table name	TRANSFILE	TRANSLATE file name
Library name		*LIBL library name
Translation table member name	TRANSMBR	ATOE3 member name

The PRTTO3X command is an interactive command that transfers PC-DOS print files from the PC to the AS/400, and adds them to the AS/400 print queue.

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

- QFNAME1** The name of the micro file to be moved to the AS/400. For additional information, see About Microcomputer Files, in Chapter 3.
- OUTQ** The output queue name where the new print item is placed when it is transferred from the micro.
- Library** The library containing the output queue. \*LIBL is the default library name.
- LINES** The number of lines per page for the new print job being added to the AS/400 print queue. The default is 66.
- LPI** The number of lines per inch to allocate to the new print job being added to the AS/400 print queue. The default is 6.
- CPI** The number of characters per inch to allocate to the new print job being added to the AS/400 print queue. The default is 10.
- WIDTH** The width of the forms being printed. Valid entries are 1 through 198. The default value is 132.

**OVERFLOW** The overflow line on the new print job being added to the AS/400 print queue. The default is 63.

**FORMTYPE** The forms type label to associate with the new print job being added to the AS/400 print queue. The default is \*STD.

**COPIES** The number of copies of the new print job to print. The default is 1.

**HOLD** Specifies whether the print file should automatically start and print the job at the completion of the translation/transfer.

**\*NO** The print file will start immediately upon termination of this command (the default).

**\*YES** The print file must be started by the operator.

**TRANSFILE** The qualified file name of the translation table file to be used. The default is TRANSLATE.

**Library** The name of the library containing the TRANSFILE. \*LIBL is the default.

**TRANSMBR** The TRANSFILE member name. The default name is ATOE3, an ASCII-to-EBCDIC table supplied with ETU.

## PRTXFRMPC Command

Parameter description	Parameter name	Parameter values
PC data file Library name	FILE	Host file name *LIBL library name
PC data member name	PCMBR	FILE member name
Output queue Library name	OUTQ	host output queue name *LIBL library name
Lines per page	LINES	66 lines per page
Lines per inch	LPI	6 lines per inch
Characters per inch	CPI	10 characters per inch
Form width	WIDTH	132 form width
Overflow line number	OVERFLOW	63 overflow line
Type of form label	FORMTYPE	*STD forms type
Number of copies	COPIES	1 number of copies
Place job on hold?	HOLD	*NO or *YES
Translation table name Library name	TRANSFILE	TRANSLATE file name *LIBL library name
Translation table member name	TRANSMBR	ATOE3 member name

The PRTXFRMPC is a batch-mode command that translates PC-DOS print files that were previously transferred from the PC by the TO3XB command. The print file is added to an AS/400 print queue after the translation.

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

**FILE** The AS/400 file name that contains the print file moved from the micro (with no translation having been done).

**Library** The name of the library containing the micro print file. \*LIBL is the default.

**PCMBR** The member containing the micro print file. The default is FILE.

**OUTQ** The output queue name, where the new print item is placed after the PRTXFRMPC translation occurs.

**Library** The library containing the output queue. \*LIBL is the default.

**LINES** The number of lines per page for the new print job being added to the AS/400 print queue. The default is 66.



<b>LPI</b>	The number of lines per inch to allocate to the new print job being added to the AS/400 print queue. The default is 6.
<b>CPI</b>	The number of characters per inch to allocate to the new print job being added to the AS/400 print queue. The default is 10.
<b>WIDTH</b>	The width of the forms being printed. Valid entries are 1 through 198. The default value is 132.
<b>OVERFLOW</b>	The overflow line on the new print job being added to the print queue (OUTQ). The default is 63.
<b>FORMTYPE</b>	The forms type label to associate with the new print job being added to the print queue (OUTQ). The default is *STD.
<b>COPIES</b>	The number of copies of the new print job to print. The default is 1.
<b>HOLD</b>	Specifies whether the print file will be held on the print queue at the completion of the translation/transfer.  <b>*NO</b> The print file will start immediately upon termination of this command (the default).  <b>*YES</b> The print file must be started by the operator.
<b>TRANSFILE</b>	The qualified file name of the translation table file to be used. The default is TRANSLATE.
<b>Library</b>	The name of the library containing the TRANSFILE. *LIBL is the default.
<b>TRANSMBR</b>	The TRANSFILE member name. The default name is ATOE3, an ASCII-to-EBCDIC table supplied with ETU.

## PRTXTOPC Command

Parameter description	Parameter name	Parameter values
PC data file Library name	PCDATA	Host file name *LIBL library name
PC data member name	PCMBR	FILE member name
Action to perform	ACTION	*COPYPRT (no default) *TRANSLATE
Spooled output file	SPLFILE	spooled output file name
Spooled output file number	SPLNBR	*LAST *ONLY spool file number
Job name or * for current job	JOB	*
User name		job name
Job number		user name
Copy file name Library name	COOYFILE	000000-999999 file name *LIBL library name
Copy file member name	COPYMBR	member name
Type of Copy file	CTLCHAR	*FCFC *PRTCTL *NONE
Lines per page	LINES	066 line number
Begin translation in which column?	FROM	001 column number
End translation in which column?	TO	132 column number
Translation table file name Library name	TRANSFILE	TRANSLATE file name *LIBL library name
Translation table member name	TRANSMBR	ATOE1 member name

The PRTXTOPC command translates a print file from the AS/400 print queue, or translates a CPYSPLF file into a format that is printable on the micro. After the print file is translated, it can be moved to the micro using the FROM3XB command.

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

**PCDATA** The AS/400 file name to contain the translated print data.

**Library** The library containing the PCDATA file. The default is \*LIBL.

**PCMBR** The PCDATA member name. The default is \*FILE (same name as PCDATA).

**ACTION** Specifies the action the program is to perform.

**\*COPYPRT** Use \*COPYPRT if the data is resident on the print queue. The command will first perform the AS/400 function CPYSPLF and then translate that file into a PC print file format. The micro print file should then be transferred to the selected micro file.

**\*TRANSLATE** use \*TRANSLATE if you have already executed the AS/400 function CPYSPLF; only the translation from the CPYSPLF format to the micro print format needs to be done at this time.

**SPFILE** Valid only on ACTION(\*COPYPRT): the AS/400 file name of the spooled output file that is to be copied to a physical file. The file name is the name of the device file that was used by the program to produce the spooled output file.

**SPLNBR** Valid only on ACTION(\*COPYPRT): the number of the spooled output file, from the job whose data records are to be copied. In addition to the spool number, the following entries are valid:

**\*LAST** Copies the highest-numbered spooled output file with the specified file name; \*LAST is the default.

**\*ONLY** Copies the spooled output file from the job, when only one has the specified file name.

**JOB** Valid only on ACTION(\*COPYPRT): the name of the job that created the spooled output file whose data records are to be copied. The default is \*, indicating that the job currently executing this command created the spooled file.

**User name** The name of the user who created the job.

**Job number** The number assigned to the job by the system.

**COPYFILE** Valid only on ACTION(\*TRANSLATE): the AS/400 physical file name of the user-created CPYSPLF file (the TOFILE name of the CPYSPLF command).

**Library** The name of library containing the COPYFILE. \*LIBL is valid.

**COPYMBR** The COPYFILE member name. The default is \*FILE (same name as for the COPYFILE parameter).

**CTLCHAR** Valid only on ACTION (\*TRANSLATE): the type of CPYSPLF file that was created: \*FCFC, \*PRTCTL, or \*NONE. See the AS/400 command CPYSPLF for additional information on formatting options.

**LINES** Valid only on ACTION (\*TRANSLATE): the number of lines per page on the spool entry(s) that was copied. The default is 66.

**FROM** The column of the printed report on which to begin translation. This can be used to restrict the copying of either sensitive data, or data that has no meaning for the

current task. The specified value must be in the range of 1 to 198. The default value is 1 (one).

**TO** The column of the printed report on which to end translation. This can be used to restrict the copying of either sensitive data, or data that has no meaning for the current task. The specified value must be in the range of 1 to 198. The default "TO" column is 132.

**TRANSFILE** The qualified file name of the translation table file to be used. The default is TRANSLATE.

**Library** The name of the library containing the TRANSFILE. \*LIBL is valid.

**TRANSMBR** The TRANSFILE member name. The default member is ETOA1, an EBCDIC-to-ASCII table supplied with ETU.

## PWRDELETEB Command

Parameter description	Parameter name	Parameter values
Qualified file name	QFNAME	micro file name
Process done on which micro?	MICLIST	*REQUESTER device name
Notify Operator?	NOTIFY	*YES or *NO
Display status message?	DSPSTMSG	*YES or *NO

Use PWRDELETEB to delete a PC file on a specified micro from either a PC in emulation mode or from a dumb terminal. Issue the PWRDELETEB command from the command line and press F4 to prompt. You can also run this command in batch mode.

The PWRDELETEB parameters are described below:

- QFNAME** The name of the micro file to be deleted.
- MICLST** The device name of the micro that contains the file to be deleted.
- \*REQUESTER** Deletes the file from the micro from which the command is initiated.
- device name** Deletes the file from the micro with this device name.
- NOTIFY** Specifies whether you will be notified if the file to be deleted is not found. Sending a message to your message queue does this.
- \*YES** You are notified if the file is not found (the default).
- \*NO** You are not notified if the file is not found.
- DSPSTMSG** Displays program status messages during the execution of the command.
- \*YES** Status messages are displayed (the default).
- \*NO** Status messages are not displayed.

## PWRFROM3XB Command

Parameter description	Parameter name	Parameter values
File name	FRMFILE	host file name translation table member name
Library name		*LIBL library name
Member name	FRMMBR	*FILE *ALL
Qualified file name	QFNAME1	micro file name
Transfer to which micro?	MICLIST	*REQUESTER device name +
Delete host file after transfer?	DLTFILE	*NO or *YES
*SAVE file type?	SAVTYP	*NO or *YES
Strip seq/date file	STRIP	*NO or *YES
Type of translation	TRANSLATE	*TEXT or *YES *NO or *NONE *DIF *BASICS *TAB
Save request definition?	SAVRQS	*NO or *YES
Request name		name of request
Translation file name	TRNSFILE	TRANLATE MACTAB file name *LIBL library name
Library name		*LIBL library name
Member name	TRANSMBR	*ETOAl member name
Record length	RECL	*FILE 0-9989
DDS or F&I file name	FLDDEFFILE	*NONE or *DDS file name *LIBL library name
Library name		*LIBL library name
Member name	FLDDEFMBR	*FILE file name
Record length	RECL	*FILE 0-9989
Field name		*ALL field name
Key field		*NONE *KEY1, *KEY2, *KEY3, ... *KEY25
Ascending or descending		*ASC *DES
Record selection	RECSEL	*NONE *IF *AND *OR *ANDIF *ORIF
Relationship		*NONE *IF *AND *OR *ANDIF *ORIF
Field to be tested		field name
Relational operator		*EQ *NE *GT *LT *LE *GE

Comparison value		numeric value or 'alphabetic value'
Entry program	ENTRYPGM	program name
Library name		*LIBL library name
Exit program	EXITPGM	program name
Library name		*LIBL library name
Display status message?	DSPSTMSG	*NO or *YES

The PWRFROM3XB command performs all functions of the FROM3XB command: it transfers files from the AS/400 to a micro and, optionally, translates the data. It also offers:

- Record and field selection -- you may transfer selected fields and records from a host file to the micro (FLDSEL, RECSEL).
- Multiple-device transfer -- a single host file may be sent to up to ten micro-computers simultaneously. The micros involved in a file transfer must be of the same platform, either DO based PC or Mac, but not both (MICLST).
- Entry and exit programs -- user-defined programs can be specified to execute before or after the transfer (ENTRYPGM, EXITPGM).
- Reusable request definitions -- The current request definition can be saved for reuse at a later time (SAVRQS).

**NOTE:** The record and field selection feature within the PWRFROM3XB command will not work properly if you define one or more additional Auxiliary Storage Pools on the AS/400. This is because ETU creates temporary work files in library QTEMP, which is always located in Auxiliary Storage Pool #1 (which comes with OS/400). If you create an additional Auxiliary Storage Pool, ETU cannot create the work files associated with the data file that is stored in any newly created Auxiliary Storage Pools.

If you do create a new Auxiliary Storage Pool, you can simulate record and field selection in a file transfer by creating a logical file or an open query file.

The parameters for the PWRFROM3XB command are described below.

**FRMFILE** The file name of the AS/400 file to be transferred to the micro.

**Library name** The name of the library containing the host file to transfer. The default, \*LIBL, searches all libraries in the user's library list.

**Member name** The FRMFILE member name.

**\*FILE** Transfers the member with the same name as the FRMFILE file (the default).

**\*ALL** Transfers all members from a qualified file. \*ALL is not valid with \*DIF translation.

**QFNAME1** The name of the micro file to contain the data from the AS/400. For additional information, see About Microcomputer Files in Chapter 3.

**MICLST** The device name of the microcomputers to which the file will be transferred.

**\*REQUESTER** Transfers the file to the micro from which the command is initiated. Use only with interactive transfers.

**device name** Transfers the file to the micro with this device name.

‡ To transfer the host file to more than one micro, enter the device name of each micro computer. (You must be authorized to use each device you specify in the MICLST parameter. If you can sign on to the device, then you are authorized to use it.) Enter a plus sign (+) and press ENTER here to display multiple prompts.

Each micro to which you transfer data must be powered on, have emulation loaded, and have the AS/400 sign-on screen displayed.

**DLTFILE** Deletes the host file after it has been transferred to the micro. \*NO is the default. To delete the file, specify \*YES. The use of this parameter is dependent on the user's authority to the host file.

**SAVTYP** Must be used to transfer files that were previously transferred to the host with the \*SAVE file type. The default is \*NO. Specify \*YES for \*SAVE file types.

**STRIP** For source physical file transfers only: Specifies whether the first 12 bytes of each record are to be ignored when reading the file. These 12 bytes contain the sequence number and date fields for source physical files. The default is \*NO. \*YES strips first 12 bytes off each record during the file transfer.

**TRANSLATE** Determines whether the data is to be translated, and if so, what type of translation is to be performed. ETU supports four translation types:

**\*TEXT** Translates the host file into ASCII text format. This is the default.

**\*DIF** Translates the host file into DIF format.

**\*BASICS** Translates the host file into BASIC Sequential format.

**\*TAB** Translates the host file into BASIC Sequential format with tab characters as the field delimiters.

**\*NONE** Indicates that no translation of the host file is to be performed. (Usually this is used only after the data is translated independently with the XLATETO command.)



**SAVRQS** Saves the current request definition so it can be reused at a later time. The request definition is this completed PWRFROM3XB command with the parameter values as you have specified them. The default is \*NO. To save the request definition, specify **\*YES** and enter the request name you desire.

**Request name** The name of the SAVRQS request definition. The request name can be up to ten characters long. (To recall the saved request, use the ETU command VWSAVRQS.)

**NOTE:** When saving a PWRFROM3XB command in which the Record Selection parameter list is used, special values should not be used for the Comparison prompt.

**TRANSFILE** The file name of the translation table file to be used when a translation is specified. The default translate file name is TRANSLATE. The file TRANSLATE is in the library containing the ETU programs.

**Mac Users:** There are two files you can use. If you use the default file (TRANSLATE), the member names have an **M** on the end of the name (for example, ETOA1M). The Mac translation members are also shipped in a file called MACTAB. The members in this file do not have the **□M□** at the end of the name (for example, ETOA1).

**Library name** The name of library containing the translation table file. The default is \*LIBL, which searches all libraries in the user's library list.

**Member name** The TRANSLATE member name. The default name is ETOA1, an EBCDIC-to-ASCII table supplied with ETU.

**FLDDEFFILE** The name of the source physical file containing the data description specifications (DDS) or RPG source file containing F and I specifications. Use the default, \*NONE, if the file being transferred was previously defined with DDS.

**Library name** The library containing the source physical file specified in FLDDEFFILE. The default, \*LIBL, searches all libraries in the user's library list.

**Member name** The source physical file member containing the DDS or F and I. The default, \*FILE, uses the member with the same name as the FLDDEFFILE source file.

**RECL** The record length of the translation work file. Valid entries are:

**\*FILE** The record length of the FRMFILE file is used.

**record** The actual record length for the translation work file.

**Length** May be from 1 to 9989 bytes long.

If an ASCII fixed-record-length translation is to be performed, specify \*FILE to produce the desired file. If not translating to ASCII fixed-length records, any record length may be specified that could allow for better performance by buffering the data.

**FLDSEL** Allows the transfer of up to 25 user-specified fields from the host to the micro. For each field to be transferred to the micro, specify the following set of parameters: Field Name, Key Field, Ascending or Descending Order.

See Chapter 9, Advanced ETU Features, for examples of using field selection.

<b>NOTE:</b> Field selection can be used only with externally described files.
--

**Field Name** The name of the field to be transferred. The default, \*ALL, transfers all the fields in the file. To transfer multiple fields, enter the name of each field to be transferred. Enter a plus sign (+) and press ENTER to display multiple prompts.

**Key Field** Sets the sort order of the records in the file after the file is transferred to the micro. The default, \*NONE, does not re-sort the records, but leaves them in sequential order.

Valid Key Field values are \*KEY1, \*KEY2, \*KEY3, up to \*KEY25. For the field with the highest sort priority, enter \*KEY1.

**Ascending or Descending** Specifies how records are sorted based on the key field value. Enter \*ASC to sort records in ascending order and \*DES in descending order

**RECSEL** Allows the transfer of user-specified records to the micro. The records that are transferred are selected based on the test cases you supply in the following parameters: Relationship, Field to be Tested, Relational Operator, and Comparison Value.

You may create up to 15 test cases. To display multiple prompts, enter a plus sign (+) and press ENTER.

An example of a test case might be: if the value in the TOTAL field is greater than 1000, then transfer the record. In this example, 'if' is the relationship, 'TOTAL' is the field to be tested, 'greater than' is the relational operator, and '1000' is the comparison value.

<b>NOTE:</b> Record selection can be used only with externally described files.
---

**Relationship** Relates one test case to another. The first test case must use \*IF. The default, \*NONE, transfers

all records in the file. The other relationships are \*AND, \*OR, \*ANDIF, \*ORIF and are used in situations where more than one test case is performed.

**Field to be Tested** Specifies the field to be tested. Use the FRMFILE parameter for the field name.

**Relational Operator** The relationship between the field to be tested and the comparison value. The operators are:

**\*EQ** - Equal to

**\*NE** - Not equal to

**\*GT** - Greater than

**\*LT** - Less than

**\*LE** - Less than or equal to

**\*GE** - Greater than or equal to

**Comparison Value** The value to which to compare the data in the field. Enter numeric values for numeric fields, alphabetic values for alphabetic fields. Enclose alphabetic fields in single quotation marks (').

<p><b>NOTE:</b> For Saved Requests, an alphabetic Comparison Value containing a space (e.g., 'red apple') is a special value.</p>
---

**ENTRYPGM** The name of the program to be run before the transfer occurs.

**Library name** The name of the library containing the specified entry program.

**EXITPGM** The name of the program to be run after the transfer occurs.

**Library name** The name of the library containing the specified exit program.

**DSPSTMSG** Displays program status messages during the execution of the PWRFROM3XB command. The default is \*YES. To turn the messages off, enter \*NO.

## PWRRENAMEB Command

Parameter description	Parameter name	Parameter values
Qualified file name	QFNAME1	micro file name
New file name	NEWFIL	new file name
Process done on which micro?	MICLIST	*REQUESTER device name
Display status message?	DSPSTMSG	*NO or *YES

Issue PWRRENAMEB on the command line and press F4 to prompt in order to rename an existing micro file on a specified micro or micros. You can issue PWRRENAMEB from a micro in emulation mode or from a dumb terminal. You can also run this command in batch mode.

The PWRRENAMEB parameters are described below:

**QFNAME** The name of the micro file to be renamed.

**NEWFIL** The new name to be given to the micro file. Do not include the path name with the file.

**MICLIST** The device name of the micro that contains the file to be renamed.

**\*REQUESTER** Renames the file on the micro from which the command is initiated.

**device name** Renames the file on the micro with this device name.

**DSPSTMSG** Displays program status messages during the execution of the command.

**\*YES** Status messages are displayed (the default).

**\*NO** Status messages are not displayed.

## PWRTO3XB Command

Parameter description	Parameter name	Parameter values
Qualified file name	QFNAME1	micro file name
Transfer from which micro?	MICLIST	*REQUESTER device name +
File name	TOFILE	host file name *MULTI
Library name		*LIBL library name
Member name	FRMMBR	*FILE *MULTI member name
Record length	RECL	*FILE 0-9989
Type of translation	TRANSLATE	*TEXT or *YES *NO or *NONE *DIF *BASICS *TAB
Add or replace records?	ADDRPL	*RELACE *ADD
File Type	FILETYPE	*DATA *SAVE *SRC *SAVF
Add seq/date field	SRCSEQ	*NO or *YES
Truncate text?	TRUNCATE	*NO or *YES
Delete micro file after transfer?	DLTFILE	*NO or *YES
Save request definition?	SAVRQS	*NO or *YES
Request name		name of request
Translation file name	TRNSFILE	TRANLATE MACTAB file name *LIBL library name *ATOE1 member name
Library name		
Member name		
DDS or F&I file name	FLDDEFFILE	*NONE or *DDS file name *LIBL library name *FILE file name
Library name		
Member name		
Entry program	ENTRYPGM	program name *LIBL library name
Library name		
Exit program	EXITPGM	program name *LIBL library name
Library name		
Display status message?	DSPSTMSG	*NO or *YES

The PWRT03XB command performs all the functions of the T03XB command: it transfers files to the AS/400 from a micro and, optionally, translates the data. In addition to these functions, PWRT03XB offers the following additional features:

- Multiple-device transfer -- a single file from one or more micros can be sent to the host. The micros involved in a file transfer must be of the same platform, either DOS-based PC or Mac, but not both.
- Entry and exit programs -- user-defined programs can be specified from within the PWRT03XB command and executed immediately before and/or after the transfer.
- Reusable request definitions -- the current request definition can be saved for reuse at a later time.

**NOTE:** Before using PWRT03XB to transfer multiple files to a single library on the host, review Transferring Files With PCFROMPC and PWRT03XB in Chapter 9 in order to minimize the risk of copying over previously transferred files.

The parameters for the PWRT03XB command are described below.

**QFNAME1** The name of the micro file to be transferred to the host. For additional information, see About Microcomputer Files in Chapter 3.

**MICLST** The device name of the source micro. Files from up to ten micros can be transferred. The file and path names must be identical on each micro.

**\*REQUESTER** Transfers the file from the micro from which the command is initiated.

**device name** Transfers the file from the micro with this device name.

‡ To transfer a micro file to multiple micros, enter the device name of each. (You must be authorized to use each device you specify in the MICLST parameter. If you can sign on to the device, then you are authorized to use it.) Enter a plus sign (+) here and press ENTER to display multiple prompts.

Each micro from which you transfer data must be powered on, have emulation loaded, and have the AS/400 sign-on screen displayed.

**TOFILE** The file name of the AS/400 file into which the micro data is to be transferred. When transferring files from multiple micros, you can transfer the files into a single host file containing multiple members (one member for each micro's file), into a single host file containing one member (the micro data is stored together in one member), or into multiple files each with one member.

**File name** Transfers the files into a single host file with one or multiple members. The name of the host file is the same as the device name of the micro

from which it was transferred.

**\*MULTI** Transfers the files into multiple host files. Data Description Specifications (DDS) must be specified when using \*MULTI as the value in the TOFILE parameter.

**Library name** The name of the library containing the host file. The default, \*LIBL, searches all libraries in your library list for the file. You must specify an existing library if the host file is to be created with this command.

**Member name** The TOFILE member name. You may use one of the following values.

**\*FILE** Transfers the micro file into a member having the same name as the host file (the default). For use with single file transfers, and in conjunction with \*MULTI as the TOFILE file name parameter.

**Member** Transfers the files into one member within the specified file.

**\*MULTI** Transfers the files into multiple members within one physical file. The members will be assigned the same name as the device name of the micro from which the data was transferred. This value is not available for \*SAVF files.

**RECL** The record length of the host TOFILE file. If the file will be allocated (created) using this command, you must include the record length of the file, or use the FLDDEFFILE parameter. If the file already exists on the host, use the default record length of zero.

**TRANSLATE** Determines if the data is to be translated, and if so, what type of translation is to be performed. ETU supports four translation types:

**\*TEXT** The qualified file resides in ASCII TEXT format and is to be translated for the AS/400. This is the default.

**\*DIF** The qualified file resides in DIF format and is to be translated for the AS/400.

**\*BASICS** The qualified file resides in BASIC Sequential format and is to be translated for the AS/400.

**\*TAB** The qualified file resides in BASIC Sequential format with tab character field delimiters and is to be translated for the AS/400.

**\*NONE** No translation is to occur. Data will reside on the host in ASCII or in the format in which it

currently exists on the micro.

**ADDRPL** For transfers to existing files: indicates whether the transferred file will replace the data in the file or be added to it. The default is \*REPLACE. To add to the existing data, specify \*ADD.

**FILETYPE** The type of TOFILE file, if the host file is to be created with this command.

**\*DATA** A physical file is created. RECL or FLDDEFFILE is used to create the data file. (This is the default.)

**\*SAVE** Permits offline storage, and moves executable programs between micros. A record length (RECL) of 1024 is required and TRANSLATE must be \*NONE. All necessary data (including Mac data and resource forks) is transferred. Data is not translated to EBCDIC, and therefore cannot be used on the host. To restore data to the micro, use the FROM3XB command with the SAVTYPE parameter.

**\*SRC** A source physical file is created. If \*SRC is used, 12 bytes will be added to the value in RECL.

**\*SAVF** A save file is created. A record length (RECL) of 528 bytes is required and TRANSLATE must be \*NONE.

<p><b>NOTE:</b> To create a *SAVF object, you must be authorized to execute the CRTSAVF command.</p>
--

**SRCSEQ** Determines whether a 12-byte source sequence number and date field is to be written in the first 12 bytes of the record. This allows the translated record to be read by the AS/400 Source Entry Utility (SEU). This parameter also affects the actual record length of the file. If you specify \*YES for this parameter, you must add 12 positions to the figure in RECL.

**\*NO** Do not add the sequence number/date field. The record length is equal to that specified (the default).

**\*YES** Add the sequence number/date field. Actual record length will be 12 greater than the record length specified in the RECL parameter.

Note that SRCSEQ can be specified even if the file is not translated, in which case untranslated data would be sequenced.

**TRUNCATE** For \*TEXT translations only: Specifies whether the text is to be truncated or folded in the event that the length of the file is longer than the record length specified in the parameter RECL.

**\*NO** The data is folded (this is the default).



**\*YES** The data is truncated, if necessary.

To truncate indicates that once the current host record is filled, any additional data for that record as it resided on the micro will be lost. Additionally, the record is written to the host only after ETU reads an end-of-record character sequence in the qualified file. The end-of-record character is defined in the translation table (see Table 8-1, position 40-41, in Chapter 8).

To fold indicates that once the current host record is filled, any additional data existing for that micro record will be written (wrapped) into the next AS/400 record.

**DLTFILE** Allows you to delete the micro file (on the micro) after it has been transferred to the host. \*NO is the default. To delete the file, specify \*YES.

**SAVRQS** Saves the current request definition so it can be used at a later time. The request definition is the completed PWRT03XB command with the parameter values as you have specified them. The default is \*NO. To save the request definition, specify \*YES and enter the request name you desire.

**Request Name** The name of the SAVRQS request definition. The request name can be up to ten characters long. (To recall the saved request, use the ETU command VWSAVRQS.)

**TRANSFILE** The file name of the translation table file to be used when translation is specified. The default translate file name is TRANSLATE. The file TRANSLATE is in the library containing the ETU programs.

**Mac Users:** There are two files you can use. If you use the default file (TRANSLATE), the member names have an **M** on the end of the name (for example, ATOE1M). The Mac translation members are also shipped in a file called MACTAB. The members in this file do not have the **M** at the end of the name (for example, ATOE1).

**Library name** The name of library containing the translation table file. The default, \*LIBL, searches all libraries in the user's library list.

**Member name** The TRANSLATE member name. The default member name is ATOE1, an ASCII-to-EBCDIC table supplied with the ETU.

**FLDDEFFILE** The source physical file containing the data description specifications (DDS) or a RPG file containing F and I specifications. Use the default, \*NONE, if you are transferring a file to an existing host file.

**Library name** The library containing the source physical file specified in the FLDDEFFILE parameter. The default, \*LIBL, searches all libraries in the user's library list.

**Member name** The source physical file member containing the DDS. The default, \*FILE, uses the member with the

same name as the source file.

**ENTRYPGM** The name of any user-specified entry program to be run prior to transferring the file.

**Library name** The name of the library containing the specified program.

**EXITPGM** The name of any user-specified exit program to be run after transferring the file.

**Library name** The name of the library containing the specified program.

**DSPSTMSG** Displays program status messages during the execution of this command. The default is \*YES. To turn the messages off, enter \*NO.

## RENAMEB Command

Parameter description	Parameter name	Parameter values
Qualified file name	QFNAME1	micro file name
New file name	NEWFIL	new file name
Display status message?	DSPSTMSG	*NO or *YES

The RENAMEB command renames a file on the micro.

**Mac Users:** The directory dialog box is unavailable for the RENAME option.

- QFNAME** The name of the micro file (including the path) to be renamed. For additional information, see About Microcomputer Files in Chapter 3.
- NEWFIL** The new name to be given to the micro file. The path of the file given in the QFNAME parameter is used.
- DSPSTMSG** Displays program status messages during the execution of the command. The default is \*YES. To turn the messages off, enter \*NO.

## SBMBATXFER Command

Parameter description	Parameter name	Parameter values
Command name to submit	SBMCMDNAME	ETU command name
Job name	JOBNAME	*JOB submitted job name
Job description	JOB	*USERPRF job description name
Library name		*LIBL library name
Job queue	JOBQ	QBATCH Job queue name
Library name		*LIBL library name
Display status message?	DSPSTSMSG	*NO or *YES

The SBMBATXFER command submits the ETU power commands (PCFROMPC, PCTOPC, PWRFROM3XB, and PWRT03XB) to batch.

**SBMCMDNAM** The name of the ETU command you want to submit to batch. Enter PCFROMPC, PCTOPC, PWRFROM3XB, or PWRT03XB.

**JOBNAME** The name of the job you are submitting to batch. The default, \*JOB, assigns the name of the job description from the user's AS/400 user profile. You can specify any name up to ten characters.

**JOB** The job description. The default, \*USRPRF, uses the job description associated with the user's AS/400 profile.

**Library name** The name of the library containing the job description. The default, \*LIBL, searches all libraries in the user's library list.

**JOBQ** The name of the AS/400 job queue to which the command will be sent. The default, QBATCH, is an IBM-supplied job queue.

**Library name** The name of the library containing the job queue. The default, \*LIBL, searches all libraries in the user's library list.

**HOLD** Specifies whether the job will be held or not once it is placed on the job queue. The default is \*NO. To hold the job, enter \*YES. You must then manually release the job.

**NOTE:** If you are submitting one of the ETU power commands to batch, you must change the default value, \*REQUESTER, in the power command to the actual device address you will use.

## SBMSCHJOB Command

Parameter description	Parameter name	Parameter values
Command	SBMCMD	command name ?
Hour in 24 hour clock format	SCHHR	00-24
Minutes	SCHMIN	00-59
Date in system format	SCHDT	*TODAY date + 1-14
Job name	JOBNAME	*JOB submitted job name
Job description	JOB	*USERPRF job description name
Library name		*LIBL library name

The SBMSCHJOB command allows you to schedule batch jobs for execution in the future. Only the ETU power commands can be scheduled using this command (PCFROMPC, PCTOPC, PWRFROM3XB, and PWRT03XB). In addition to the power commands, you can also schedule any non-ETU commands that can be submitted to batch.

### Before You Begin

Before you can schedule a job, you must first start the job called JOBSCHED, which controls the job scheduling. This job is part of a procedure that submits a program called SCHPGM to the QINTER subsystem. Every time QINTER is shut down, the JOBSCHED job must be restarted or the job scheduling program will not work.

To restart JOBSCHED, enter the following command string on the AS/400 command line:

```
sbmjob job(jobsched) jobd(****) rqsdt('call schpgmc')  
jobq(qinter) inqmsgrpy(*sysrpy1)
```

where \*\*\*\* is the job description that a job submitted to batch with the SBMSCHJOB command will use for its operating parameters. These operating parameters include the library list and output queue to be used.

The time you specify for the SBMSCHJOB command is the time the job will be released; not necessarily the time the job will be processed. The time at which the job is processed depends on the number of jobs ahead of it in the job queue.

## To Schedule a Job

To schedule a job, first start a job called SCHPGM. This job controls the SBMSCHJOB command. Then enter SBMSCHJOB on the AS/400 command line, press F4, and fill out the following parameters.

**SCHCMD** The name of the command you want to schedule. You can enter the command followed by the entire command string, or enter a question mark (?) followed by the command name, which displays the command prompt after you complete the remaining parameters.

**SCHHR** The hour of the day you want the job to be released to batch. Enter the time in 24-hour (military) format.

**SCHMN** The minute of the hour. The default is 00. Valid entries are 00 through 59.

For example, to release a job at 3:30 pm, enter 15 as the SCHHR, and 30 as the SCHMN.

**SCHDT** The date, in host system format, that you want the job released. The default is \*TODAY, which releases the job on the current day at the specified time.

For instance, if the system format is DDMMYY, enter the 041591 for April 15, 1991.

Another way to specify a date in the future is to enter a plus sign (+) followed by a value from 1 to 14. This specifies the number of days from the current day to release the job.

**SCHJOB** The job name. The default is \*JOBID which uses the name of the user's job description for the job name. You can use any name for the job (up to ten characters).

**SCHJOBID** The job description. The default is QBATCH, an IBM-supplied job description that uses the job queue of the same name.

**Library** The library containing the job description. \*LIBL is the default.

**NOTE:** The time you specify for the SBMSCHJOB command is the time the job will be released, but not necessarily the time the job will be processed. The time at which the job is processed depends on the number of jobs ahead of it in the job queue.

## STRSEU Command

Parameter description	Parameter name	Parameter values
Source file	SRCFILE	TRANSSRC *PRV file name
Library name		*LIBL library name
Source member	SRCMBR	*SELECT *PRV source member file name
Source type to create	TYPE	*SAME type
Text description	TEXT	*BLANK descriptive text

The STRSEU (Start Source Entry Utility) command is the AS/400 command used to change or create a translation table. This command loads the Source Entry Utility (SEU), which creates, changes, displays, or prints a source member. SEU is part of the IBM AS/400 Application Development System. For information on SEU, refer to the IBM manual AS/400 Application Tools SEU Reference Manual and User's Guide SC21-7722.

**SRCFILE** The name of the source file to be worked with.

**Library** The name of the library containing the SRCFILE file (the same library in which ETU is stored). \*LIBL is the default.

**SRCMBR** The name of the source physical file member to be edited or created. The default value depends on whether you specify the SRCFILE parameter.

**\*PRV** The default if you do not specify the SRCFILE parameter. \*PRV selects the name of the previous source member used.

**\*SELECT** The default if you do specify a value in the SRCFILE parameter. Select a member to edit, display, print, or remove.

### source file member name

The name of the member you are creating or editing. It is recommended that a new member be created for update, copying one of the originally supplied members:

#### PC-DOS Members

**ETOA1** EBCDIC to ASCII (variable length)

**ETOA2** EBCDIC to ASCII (fixed length)

**ATOE1** ASCII to EBCDIC (variable and fixed length)

**ATOE3** ASCII to EBCDIC (print translation)

#### Mac Members

**ETOA1M** EBCDIC to ASCII (variable length)

**ETOA2M** EBCDIC to ASCII (fixed length)

**ATOE1M** ASCII to EBCDIC

**TYPE** The type of source to be edited or created. Specify any combination up to 10 characters, or one of the AS/400 types supported by SEU.

**\*SAME** Specifies that the same source type is used as when the member was edited.

**type** \*BAS, \*BASP, \*CBL, \*CL, \*CMD, \*DSPF, \*LF,  
\*PF, \*PLI, \*PRTF, \*PRG, \*RPT, \*TXT, \*CLP,  
\*CMNF, \*DFU, \*QRY, \*MXDF

**TEXT** You can enter a description that identifies the source member. The description can be up to 50 characters. The default is \*BLANK.



## TESTB Command

Parameter description	Parameter name	Parameter values
Qualified file name	QFNAME	micro file name
Notify operator?	NOTIFY	*YES *NO
Starting position in LDA	POS	no default 1-1012
Display status messages?	DSPSTMSG	*YES *NO

The TESTB command tests for the existence of a file on the micro. The result of the test is returned in a data area (PCTDTAARA) in QTEMP for later use. If you specify NOTIFY (\*YES), you will be notified if the file exists via a completion message.

**QFNAME** The name of the micro file to be tested. For additional information, see About Microcomputer Files in Chapter 3.

**NOTIFY** Determines whether a message about the existence of the file is to be sent to the display

**\*YES** Always send message (default).

**\*NO** Do not send any message.

**\*LDA** Do not send any message but put result in LDA (convenience to S/36 users).

**POS** The starting position in the LDA for the result. Use this parameter only with NOTIFY (\*LDA).

**DSPSTMSG** Displays program status messages during the execution of the command. The default is \*YES. To turn the messages off, enter \*NO.

## TO3XB Command

Parameter description	Parameter name	Parameter values
File name	TOFILE	host file name
Library name		*MULTI *LIBL library name
Member name	FRMMBR	*FILE *MULTI member name
Record length	RECL	*FILE 0-9989
File Type	FILETYPE	*DATA *SAVE *SRC *SAVF
Add seq/date field	SRCSEQ	*NO or *YES
Type of translation	TRANSLATE	*TEXT or *YES *DIF *BASICS *TAB *NO or *NONE
Translation file name	TRNSFILE	TRANLATE MACTAB file name
Library name		*LIBL library name
Member name	TRANSMBR	*ATOE1 member name
Truncate text	TRUNCATE	*NO *YES
DDS or F&I file name	FLDDEFFILE	*NONE or *DDS file name
Library name		*LIBL library name
Member name	FLDDEFMBR	*FILE file name
Qualified file name	QFNAME1	micro file name
Save request definition?	SAVRQS	*NO or *YES
Request name		name of request
Display status message?	DSPSTMSG	*NO or *YES

The TO3XB command transfers a micro file to the AS/400, and, optionally, translates the data. The data is translated according to the specified translation type and translation table. Four different types of translation can be selected: \*DIF, \*BASICS, \*TAB, and \*TEXT (see Chapter 4, Learning About ETU, for more information on each type), or \*NONE for no translation.

**NOTE:** \*DIF, \*BASICS, and \*TAB translations require that the fields in the IBM data file be defined in advance of the file transfer. The fields may be defined with data description specifications (DDS) associated with the data file, or with a source member created with standard RPG F and I specifications that describe the field layouts of the data file.

**TOFILE** The qualified file name of the AS/400 file into which the micro data is to be written.

**Library** The name of the library containing the TOFILE. The default is \*LIBL. You must specify an existing library

name if the file is to be created with the TO3XB command.

**MBR** The TOFILE member name. If the member already exists, it will be cleared. If the member does not exist, it will be added to the file. \*FILE indicates that the member name is the same as the value in the TOFILE parameter.

**RECL** The record length of the TOFILE file to be used in either allocating (creating) the file or verifying the record length of an existing file. A record length of zero can be specified if the file already exists.

**FILETYPE** The type of AS/400 file to be created (if necessary).

**\*DATA** A physical file is created. The value entered in RECL is used to create the data file. This is the default.

**\*SAVE** Permits offline storage of micro files, and moves executable programs between micros. Requires a RECL value of 1024 bytes. All necessary data (including Mac data and resource forks) is transferred. Data is not translated to EBCDIC, and therefore cannot be used on the host. To restore data to the micro, use the FROM3XB command with the SAVTYPE parameter and TRANSLATE(\*NONE).

**\*SRC** A source physical file is created. If a \*SRC file is created, 12 bytes will be added to the RECL value.

A source physical file can be read by SEU.

**\*SAVF** A save file is created. A RECL value of 528 bytes and TRANSLATE(\*NONE) is required.

<p><b>NOTE:</b> To create a *SAVF object, you must be authorized to execute the CRTSAVF command, normally restricted to the security officer. If the *SAVF object already exists, no special authority is required.</p>
---

**SRCSEQ** Determines whether a 12-byte source sequence number and date field is to be written in the first 12 bytes of the record. This allows the translated record to be read by the S/38 Source Entry Utility (SEU). This parameter also affects the actual record length of the file. If you specify \*YES for this parameter, you must add 12 positions to the figure in RECL.

**\*NO** Do not add the sequence number/date field. The record length is equal to that specified (the default).

**\*YES** Add the sequence number/date field. Actual record length will be 12 greater than the record length specified in the RECL parameter.

Note that SRCSEQ can be specified even if the file is not translated, in which case untranslated data would be sequenced.

**TRANSLATE** Determines whether the data is to be translated, and if so, what type of translation is to be performed.

**\*TEXT** or **\*YES** The qualified file resides in ASCII Text format and is to be translated for the AS/400. This is the default.

**\*DIF** The qualified file resides in DIF format and is to be translated for the AS/400.

**\*BASICS** The qualified file resides in BASIC Sequential format and is to be translated for the AS/400.

**\*TAB** The qualified file resides in BASIC Sequential format with tab character field delimiters and is to be translated for the AS/400.

**\*NO** or **\*NONE** No translation is to be performed. After the file is transferred to the host, it will still reside in the original ASCII format or in the format that currently exists on the micro.

**TRANSFILE** The qualified file name of the translation table file to be used when a translation is specified. The default translate file name is TRANSLATE.

**Mac Users:** There are two files you can use. If you use the default file (TRANSLATE), the member names have an **M** on the end of the name (for example, ATOE1M). The Mac translation members are also shipped in a file called MACTAB. The members in this file do not have the **M** at the end of the name (for example, ATOE1).

**Library** The library containing the TRANSFILE. \*LIBL is a the default.

**TRANSMBR** The TRANSFILE member name. The default name is ATOE1, an ASCII-to-EBCDIC table supplied with ETU.

**TRUNCATE** For \*TEXT translations only: Specifies whether the text is to be truncated.

**\*NO** The data is folded (this is the default).

**\*YES** The data is truncated, if necessary. To truncate indicates that once the current host record is filled, any additional data for that record as it resided on the micro will be lost. Additionally, the record is written to the host only after ETU reads an end-of-record character sequence in the qualified file. The end-of-record character is defined in the translation table (see Table 8-1, position 40-41, in Chapter 8).

To fold indicates that once the current host record is filled, any additional data existing for that micro record will be written (wrapped) into the next AS/400 record..

**FLDDEFFILE** When transferring to a new file, specify a source physical file containing either physical file data description specifications (DDS), or an RPG source file containing F and I specifications.

Use the default, \*NONE, or its equivalent, \*DDS, for existing host files.

- FLDDEFMBR** The member contained in the FLDDEFFILE. The default, \*FILE, uses the member with the same name as the file specified in FLDDEFFILE.
- QFNAME1** The name of the micro file to be transferred. For additional information, see About Microcomputer Files, in Chapter 3.
- SAVRQS** Saves the current request definition so it can be used at a later time. The request definition is this completed TO3XB command with the parameter values as you have specified them. The default is \*NO. To save the request definition, specify \*YES and enter the request name you desire.
- Request Name** The name of the SAVARQS request definition. The request name can be up to ten character long. (To recall the saved request, use the ETU command VWSAVRQS.)
- DSPSTMSG** Displays program status messages during the execution of the command. The default is \*YES. To turn the messages off, enter \*NO.

## VWSAVRQS Command

The VWSAVRQS command displays the request definitions saved with the FROM3XB, PWRFROM3XB, PWRT03XB, PCTOPC, PCFROMPC, and TO3XB commands. Once you display the request definitions, you can run them interactively or submit the power commands to batch. You can also change or delete any of the saved requests. You may take action on more than one saved request at a time.

To use the command, enter VWSAVRQS on the AS/400 command line. This displays all previously saved requests made by the user executing the VWSAVRQS command.

To take action on a request, enter one of the following codes adjacent to it.

The option codes and the actions they represent are:

- |                          |   |
|--------------------------|---|
| <b>1 Execute</b>         | Interactively runs the command using the saved definition.  |
| <b>2 Submit to batch</b> | Prompts for the job description parameters and submits the request to batch. Only the ETU power commands may be submitted to batch; not the FROM3XB or TO3XB commands. Batch command cannot be longer than 256 characters.                    |
| <b>3 Change</b>          | Allows you to change the saved request.   |
| <b>4 Delete</b>          | Deletes the specified request   |
| <b>5 Extended view</b>   | Displays the full request, if it is longer than 45 characters. Only 45 characters are normally displayed on the screen. If the saved request is longer than 45 characters, it will be displayed on this screen followed by an ellipsis (...). |

### VWSAVRQS Notes

Fold/Unfold: The function key F1 provides an alternative view of the highlighted request. It shows you the name of the user who created the definition, the device from which it was created, the creation date, and creation time. It also lists the name of the user who last changed the definition, the workstation from which it was changed and the date and time that the change occurred.

Figure 10-3 VWSAVRQS prompt screen

```
1/07/03          View saved request definitions

Options:  1= Execute          2= Submit to batch  3= Change
          4= Delete           5= Extended view

Rqs name   Cmd name   Cmd parms
          PWRT03XB   QFNAME1('c:\bootlog.txt') M1CLST(NLRISCLIF...

F1= Unfold/Fold      F3= Exit
```



## XLATEFROM Command

Parameter description	Parameter name	Parameter values
PC data file name Library name	PCDATA	PC data file on the host *LIBL library name
PC member name	PCMBR	*FILE member name
Host output data file Library name	TOFILE	host file name *LIBL library name
Host output file member name	MBR	*FILE host member name
Record length	RECL	*FILE 0-9989
File Type	FILETYPE	*DATA *SRC
Add seq/date field	SRCSEQ	*NO or *YES
Type of translation	TRANSLATE	*TEXT or *YES *DIF *BASICS *TAB
Translation file name  Library name  Member name	TRNSFILE	TRANLATE MACTAB File name *LIBL library name
Truncate text	TRUNCATE	*ATOE1 member name
Job description Library name	TRUNCATE	*NO *YES
Field translation definition	JOB	job description *LIBL library name
DDS or F&I file name Library name	FLDDEFFILE	*DDS file name
Member name	FLDDEFMBR	*LIBL library name *FILE *FIRST *LAST member name

The XLATEFROM command translates a file that was received from the micro via the TO3XB command. It is used where it would be advantageous to perform the translation process at a later time (for instance, on a long-distance line where connection time is being kept to a minimum).

This command can be put in a job queue by using the SBMJOB command (in that case, do not specify the JOB parameter). This command can be executed from any terminal on the host system, not necessarily from the connecting micro. To use XLATEFROM, you must have executed the TO3XB command with the FILETYPE, SRCSEQ, and TRANSLATE parameters coded as follows:

**filetype(\*data) srcseq(\*no) translate(\*no)**

**PCDATA** The (host) file name containing the untranslated micro data.



**Library** The library containing the PCDATA file. \*LIBL is the default.

**PCMBR** The PCDATA file member name. The default is \*FILE.

**TOFILE** The qualified file name of the AS/400 file into which the micro data is to be written.

**Library** The library containing the TOFILE file. \*LIBL is the default.

**TOMBR** The TOFILE member name. If the member does not exist, it will be added to the file. If it does exist, the member will first be cleared before the current data is written. The default is \*FILE.

**RECL** The record length of the AS/400 file. It is the actual data length. If you specify SRCSEQ(\*YES), the record length used in allocating the file is 12 greater than that supplied in this parameter, to make room for the sequence number/date field being added. You must specify the record length of the file if the TOFILE file is to be created. If the file already exists, the specified length must be 0 or must match the file record length. Any length from 1 to 9989 is supported.

**FILETYPE** The type of TOFILE file to be created.

**\*DATA** A physical file is created (the default).

**\*SRC** A source physical file is created.

**SRCSEQ** Specifies whether a 12-byte source sequence number and data field is to be written in the first 12 bytes of the record. This allows the translated record to be read by the AS/400 Source Entry Utility (SEU). Note that SRCSEQ can be specified even if the file is not translated, in which case untranslated data would be sequenced. This parameter also affects the actual record length of the file.

**\*NO** Do not add the sequence number/date field. The record length is equal to that specified. \*NO is the default.

**\*YES** Add the sequence number/date field. Actual record length is 12 greater than the specified record length.

**TRANSLATE** The type of file translation that is to occur.

**\*TEXT** The qualified file exists in ASCII text format. This is the default.

**\*DIF** The qualified file exists in DIF format.

**\*BASICS** The qualified file exists in BASIC Sequential format.

**\*TAB** The qualified file exists in BASIC Sequential format with tab characters as field delimiters.

**TRANSFILE** The qualified file name of the translate table file to be used. The default is TRANSLATE.

**Library** The library name containing the TRANSFILE file. \*LIB is the default.

**Mac Users:** There are two files you can use. If you use the default file (TRANSLATE), the member names have an **M** on the end of the name (for example, ATOE1M). The Mac translation members are also shipped in a file called MACTAB. The members in this file do not have the **M** at the end of the name (for example, ATOE1).

**TRANSMBR** The TRANSFILE member name. ATOE1, an ASCII-to-EBCDIC table, is the default.

**TRUNCATE** For \*TEXT translations only: Specifies whether the text is to be truncated.

**\*NO** The data is folded (this is the default).

**\*YES** The data is truncated, if necessary.

To truncate indicates that once the current host record is filled, any additional data for that record as it resided on the micro will be lost. Additionally, the record is written to the host only after ETU reads an end-of-record character sequence in the qualified file. The end-of-record character is defined in the translation table (see Table 8-1, position 40-41, in Chapter 8).

To fold indicates that once the current host record is filled, any additional data existing for that micro record will be written (wrapped) into the next AS/400 record.

**JOB** The name of the job description to be used if XLATEFROM is to be placed in a job queue. If left blank, the XLATEFROM command is executed interactively (this is the default). When JOB is specified, XLATEFROM will execute SBMJOB command automatically, and then release the micro for other work.

**Library** The library name containing the job description. \*LIBL is the default.

**FLDDEFFILE** When transferring to a new file, specify either a source physical file containing physical file data description specifications (DDS), or RPG source file containing F and I specifications.

Use the default, \*NONE, when translating to existing host files.

**Library** The name of the library containing the FLDDEFFILE file. \*LIBL is the default.

**FLDDEFMBR** The FLDDEFFILE member name. You can enter the member name or one of the following values:

**\*FILE** Transfers the member with the same name as the FLDDEFFILE (the default value).

**\*FIRST** Transfers the first member (arrival sequence) in the file.

**\*LAST** Transfers the last member (arrival sequence) in the file.

## XLATETO Command

Parameter description	Parameter name	Parameter values
Host data file name	FRMFILE	host file name containing the untranslated data
Library name		*LIBL library name
Host data member name	FRMMBR	*FILE *ALL *FIRST *LAST
PC data file for use by PC	PCDATA	host file name to contain the translated data
Library name		*LIBL library name
Host member name	PCMBR	*FILE member name of PCDATA file
Add seq/date field	SRCSEQ	*NO or *YES
Type of translation	TRANSLATE	*TEXT or *YES *DIF *BASICS *TAB
Translation file name	TRNSFILE	TRANLATE MACTAB
Library name		*LIBL library name
Translation table member name	TRANSMBR	*ETOAl member name
Record length	RECL	*FILE 0-9989
File Type	FILETYPE	*DATA *SRC
Job description	JOB	(blank) job description
Library name		*LIBL library name
Field translation definition	FLDDEFFILE	*NONE or *DDS file name
Library name		*LIBL library name
Member name	FLDDEFMBR	*FILE *FIRST *LAST member name

The XLATETO command translates an AS/400 physical file member prior to transferring it to a micro. It is intended for use in instances where translation is to occur sometime before the actual transfer with the FROM3XB command (for instance, on a long-distance telephone line in order to keep line costs down). This command can be put on the JOBQ with the SBMJOB command (in that case, do not specify the JOBD parameter). This command may be executed from any terminal on the AS/400, not necessarily the connecting micro, as it does not involve any interaction with the micro.

When using FROM3XB to transfer the resulting data, code the FROM3XB parameters STRIP and TRANSLATE as follows:

**strip(\*no), translate(\*no)**

**FRMFILE** The file name of the AS/400 file to be translated.

**Library** The library containing the FRMFILE file. \*LIBL is the default, specifying that all libraries in the library list are to be searched.

**FRMMBR** The FRMFILE member name. You can enter the member name or one of the following values:

**\*FILE** Translates the member with the same name as the FRMFILE file (the default).

**\*ALL** Translates all members in the file into one member. \*ALL is not valid with \*DIF translation.

**\*FIRST** Processes the first member (arrival sequence) in the file.

**\*LAST** Processes the last member (arrival sequence) in the file.

**PCDATA** The name of the AS/400 physical file that is to contain the translated data.

**Library** The library containing the PCDATA. \*LIBL is the default, specifying that all libraries in the library list are to be searched.

**PCMBR** The PCDATA file member name.

**STRIP** For source physical files only: Specifies whether the 12-byte source sequence number and data field for source physical files is to be removed before translation.

**\*YES** Strip off (ignore) the first 12 bytes.

**\*NO** Do not strip off any data from the record (default).

**TRANSLATE** Specifies the type of file translation that is to be performed:

**\*TEXT** Converts the file to text format (the default).

**\*DIF** Converts the file to DIF format.

**\*BASICS** Converts the file to BASIC Sequential format.

**\*TAB** Converts the file to BASIC Sequential format with tab characters as the delimiters.

**TRANSFILE** The file name of the translation table file to be used. The default is TRANSLATE.

Mac Users: There are two files you can use.

If you use the default file (TRANSLATE), the member names have an **M** on the end of the name (for example, ETOA1M). The Mac translation members are also shipped in a file called MACTAB. The members in this file do not have the **M** at the end of the name (for example, ETOA1)

**Library** The library containing the TRANSFILE. \*LIBL is the default

**TRANSMBR** The TRANSFILE member name. The default is ETOA1.

**RECL** The record length of the translation work file record length. Valid record length entries are:

**\*FILE** Indicates that the record length of the FRMFILE will be used.

**0** Specifies that PCDATA file already exists and the current record length will be used.

**1-9989** Specifies the record length for the translation work file.

If an ASCII fixed-record-length translation is to be performed, specify \*FILE to produce the desired file. If you are not translating to ASCII fixed-length records, you may specify any record length. This could allow for better performance by buffering the data.

**JOB** The qualified name of the job description to be used if XLATETO is to be placed in a job queue. If blank, the XLATETO command will be executed interactively (this is the default). When JOB is specified, XLATETO will execute the SBMJOB command automatically, and then release the micro for other work.

**Library** The library containing the job description. \*LIBL is the default.

**FLDDEFFILE** If the file being translated is the translation type \*DIF, \*BASICS, or \*TAB, and is not externally defined with either DDS or F&I specs, you must specify either a source physical file containing data definition specifications (DDS), or an RPG source file containing F&I specifications.

When transferring a \*TEXT file or any other file that was previously defined with DDS or F&I specs, use the default, \*NONE, or its equivalent, \*DDS.

**FLDDEFMBR** The FLDDEFFILE member name. You can enter the member name or one of the following values:

**\*FILE** - Transfers the member with the same name as the FLDDEFFILE file (the default value).

**\*FIRST** Transfers the first member (arrival sequence) in the file.

**\*LAST** Transfers the last member (arrival sequence) in the file.

## *Part Four*

# Appendices

The "Appendices" provide reference and supplemental information.

<b>A-1</b>	<b>Inside ETU</b>
<b>B-1</b>	<b>ETU Compatibility</b>
<b>C-1</b>	<b>Creating DDS Specifications</b>
<b>D-1</b>	<b>Creating F &amp; I Specifications</b>
<b>E-1</b>	<b>ETU Error Messages</b>
<b>F-1</b>	<b>EBCDIC/ASCII Printable Characters</b>
<b>G-1</b>	<b>System/36-mode Commands</b>
<b>H-1</b>	<b>International Translational Tables</b>
<b>I-1</b>	<b>ETU Automation</b>
<b>J-1</b>	<b>ETU Throughput Considerations</b>

**GLOSSARY**

**INDEX**

# INSIDE ETU

About This Chapter

Editing User-defined Menu Options

EDTUDF Parameters

EDTUDF Example

Configuring the Audit Trail

The CFGAUDIT Parameters

Checking the Audit Trail

Working With ETU Security

Granting Authority to a Command

Revoking Authority From a Command

Regranting Authority to a Previously Revoked Command

Securing Devices on the AS/400

Removing ETU From the AS/400



## About This Chapter

ETU includes utilities that allow you to customize the way ETU works on your AS/400. This chapter describes the utilities and explains how to use them. This chapter is intended for experienced host users and the security officer.

Some tasks you can perform with the ETU utilities include:

- adding user-defined menu options to the ETU menu;
- changing how the ETU audit trail is configured;
- granting and revoking authority to the ETU commands;
- removing ETU from your AS/400.

## Editing User-defined Menu Options

With ETU it's possible to add menu options, called user-defined functions, to the ETU Main menu. A user-defined function is any command or program that can be run on the AS/400. Even ETU utilities, like EDTUDF, can be added to the ETU menu. Once a user-defined function has been added to the menu, you can execute the option as you do any ETU option -- by entering the option number on the ETU command line and pressing ENTER.

To add a user-defined function to the menu, follow the instructions below:

1. If you adding a user-defined function for your own menu, enter the command EDTUDF on the AS/400 command line and press Enter. Then skip to step 3.

If you are adding a user-defined function for a user other than yourself, press F4.

You are prompted to enter a user profile. You can add user-defined functions for a single user or a group of users. To add a user-defined function for a single user, enter the user's profile name. To add a user-defined function for a group of users, enter the group profile name.

2. Press ENTER.

The Edit User-defined Functions screen displays. The first time you add a user-defined function, the utility is in add mode. Each time thereafter it is in change mode. To switch from change mode to add mode, press F9.

3. On the Edit User-defined Functions screen, supply values for the Description, Function, and Mode parameters. These parameters are described below.
4. After you have completed adding the user-defined option, press F3 to return to the Main menu.

**NOTE:** In order for the new menu option to display on the ETU Main menu, you must exit the ETU menu (press F3) and re-display the ETU menu by issuing the WSMENUS command.

## EDTUDF Parameters

<b>Description</b>	The description of the user-defined function as it will appear on the ETU Main menu. It may be up to 32 character long.
<b>Function</b>	The AS/400 command that will run when the menu option is issued. Only the first 20 characters will be displayed.
<b>Mode</b>	<p>Specifies how the command will execute when selected from the ETU Main menu. Valid entries are:</p> <p><b>*PMT</b> Displays the prompt screen for the command specified in the Function parameter.</p> <p>Before the user-defined function can be executed, you must define the command request by entering values into the appropriate parameters.</p> <p><b>*EXE</b> Executes the user-defined option immediately upon selection of the menu option. You will pre-define the command request while using this EDTUDF utility. Whenever you select the user-defined option from the menu, the pre-defined command request will execute.</p>

## EDTUDF Example

In this example, you will add the ETU utility EDTUDF to the ETU Main menu. Define the menu option so that you are prompted for parameters each time you call the option from the menu so it displays only on your own menu, and appears on your menu as Edit User-defined Functions.

1. Enter the EDTUDF command on the AS/400 command line.

**edtudf**

2. Press F4 to prompt for the parameters.

The Enter User Profile screen displays.

3. Since you are adding this menu option only to your own menu, use the default \*USER.

Press ENTER.

The Edit User Defined Function screen displays.

4. At the Description prompt, enter the name of the option as you want it to appear on the menu (including UPPER and lower case characters):

**Edit User-defined Functions**

5. At the Functions prompt, enter the name of the command you are adding to the menu:

**edtudf**

6. At the Mode prompt, enter:

**\*pmt**

If you had entered \*exe, the EDTUDF prompt screen would have displayed and you would have been expected to define the request as you would have liked it to execute each time the option is chosen from the menu.

7. Press Enter to add the user-defined function.
8. Press F3 to end the command and return to the Main ETU menu.

For more information on EDTUDF, see Chapter 10, ETU Command Reference.

## Configuring the Audit Trail

An audit trail is produced for the ETU commands FROM3XB, PCFROMPC, PCTOPC, PWRFROM3XB, PWRT03XB, and TO3XB. The purpose of an audit trail is to inform you whether a file transfer operation was successful, and, if not, why it was not.

When you first install ETU, the audit trail is configured as active with entries sent on hold to the QPRINT output queue. By re-configuring the audit trail, you can turn off the audit trail, send the audit trail entries to an output queue of your choice, or send the entries directly to a printer, rather than put them on hold.

To change the audit trail configuration, display the Configure Audit Trail prompt screen by entering the CFGAUDIT command on the AS/400 command line and prompting (press F4).

You may change the audit trail configuration by changing the values of the parameters described below.

Figure A-1 The CFGAUDIT prompt screen

```
Configure audit trail: (CFGAUDIT)

Type choices, press Enter.

Create audit trail: . . . . . *YES          *YES, *NO
Audit trail output queue: . . . QPRINT      Name
Library name: . . . . . *LIBL             Name, *LIBL
Hold spool file: . . . . . *YES          *YES, *NO

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

## The CFGAUDIT Parameters

- CRTAUDIT** Turns the audit trail on or off. The default, \*YES, creates an audit trail for all users on the system. To turn off the audit trail for all users, enter \*NO.
- OUTQ** The name of the output queue to which the audit trail entries are sent. The default is QPRINT. You may enter any valid AS/400 output queue.
- Library** The library in which the output queue resides. \*LIBL is a valid library name.
- HOLD** Determines whether the audit trail entries are placed on the output queue on hold status or are released to the printer. The default, \*YES, places the entries on hold. To have the entries go directly to a printer, enter \*NO.

## Checking the Audit Trail

All ETU transfer operations performed with the above-named ETU commands create entries in the audit trail in the QPRINT output queue. To view the audit trail, display the QPRINT output queue by entering the command WRKOUTQ QPRINT on the AS/400 command line. Each ETU audit trail entry is listed in the output queue as ETUADTPP.

To view an entry, enter 5 next to the entry and press ENTER. The audit trail for the entry you selected displays. The audit trail includes the following information:

- Command** Name of the transfer command.
- File name** The qualified file name of the file that was transferred.
- Device** The name of the device or devices from which or to which the file was transferred.
- Status** A two-digit code describing the status of the file transfer. Possible codes are listed in Table A-1.

Table A-2 The ETU Audit Trail Status Messages

Status Code	Status Message
00	Transfer Successful
01	Unsuccessful Not able to acquire device
02	Unsuccessful Invalid return code from device during transfer
03	Unsuccessful Invalid return code from device during decode
04	Unsuccessful Invalid return code from device during allocate
AA	Unsuccessful Device Error

**Description** A description of the file transfer operation.

## Working with ETU Security

For the Security Officer: The ETU menuing system takes advantage of the security provided with the AS/400. This allows you to decide which users on your system will have access to which ETU commands. When ETU is first installed on your host system, the only profiles that are authorized to the ETU power commands are QSECOFR, and any other user profile with all object (\*ALLOBJ) authority.

The menu options for the power commands do not display on the ETU menus of other users. To allow a user to have access to a power command, follow the instructions below in Granting Authority to a Command.

All ETU commands, with the exception of the power commands, are shipped with \*PUBLIC object authority and are available to all users. If you do not want a user to have access to an ETU command, use the GRTOBJAUT command (grant object authority) with \*EXCLUDE as the value for the Authority parameter.

**NOTE:** If a user profile contains special authority (SPCAUT) value \*ALLOBJ, all users with that profile automatically have access to the ETU power commands on the ETU Main menu. To revoke the power commands from this user profile, you must change the user profile, removing \*ALLOBJ from SPCAUT value.

Systems with security level of 10 or 20 will automatically have \*ALLOBJ in SPCAUT.

## Granting Authority to a Command

This section describes how to grant and revoke authority to the ETU commands. When you grant authority to a power command, or any command for that matter, you can grant authority for a single user or a group of users. Once authority to a command has been granted for a user, the command will display on the user's ETU menu.

To grant authority to a power command:

1. Sign on to the AS/400 as QSECOFR.
2. Enter the GRTOBJAUT command (grant object authority) on the command line and prompt (press F4):

**grtobjaut**

3. Enter the name of the command to which you wish to grant authority.
4. Explicitly grant authority to each user who will be authorized to use the command.

It is not necessary to grant authority to commands other than the power commands. These commands can be used by all users because of their \*PUBLIC object authority.



## Revoking Authority From a Command

When you revoke authority to a command, the users from whom the command is revoked will no longer be able to use the command, nor will the command's menu option display on ETU menus. How you revoke authority depends on the command you are revoking.

To revoke authority from an ETU power command:

1. Sign on to the AS/400 as QSECOFR.
2. Enter the RVKOBJAUT command (revoke object authority) on the command line and prompt (press F4):

**rvkobjaut**

3. Enter the name of the command for which you wish to revoke authority.
4. Enter the user or profile name for which you wish to revoke authority.
5. Change the authority parameter to \*ALL.
6. Press ENTER.

To revoke authority from an ETU command other than a power command, issue the GRTOBJAUT command with \*EXCLUDE as the value for the AUTHORITY parameter.

## Regranting Authority to a Previously Revoked Command

How you regrant authority to an ETU command that has previously been revoked depends on whether the command is a power command or not.

To regrant authority to a power command, use the GRTOBJAUT command just as you would when first granting authority.

To regrant authority to a non-power command, use the RVKOBJAUT command with \*EXCLUDE as the value in the AUTHORITY parameter.

## Securing Devices on the AS/400

For the Security Officer: The ETU power commands allow a properly authorized user to transfer files to and from multiple devices. The devices to which a user is authorized can be limited using the security built into the AS/400. Then, if a user tries to transfer data to or from the device to which he is not authorized, the program ends and sends the following error message to the screen:

**Unable to open display file with device specified.**

In order to take advantage of device-specific authority, your AS/400 must be configured for a security level of 30 or above. If your system is configured below level 30, all users have the authority to access all devices.

To exclude a device from a user, follow standard system protocol for limiting device authority on the AS/400. For more information about securing devices, see AS/400 Security Concepts and Planning, SC21-8083.

## Removing ETU From the AS/400

ETU includes a utility that makes it easy to remove the ETU software from your host system. (You may need to remove ETU from your system if, in the future, you upgrade to a new version of ETU.) This utility, called RMVETU, removes all the ETU objects from the library in which you loaded ETU, and leaves intact all non-ETU objects in that library.

**NOTE:** It is critical that you backup your ETU library prior to running this command, in the event that any user data located in this library is inadvertently removed.

To remove ETU from your AS/400, follow the steps below:

1. Sign on to the AS/400 as QSECOFR.
2. Enter the RMVETU command on the AS/400 command line and prompt (press F4):

**rmvetu**

The RMVETU prompt screen displays.

3. Enter the name of the AS/400 library containing the ETU programs.
4. Press ENTER.

The ETU objects stored in the specified library are deleted.

# ETU COMPATIBILITY

About this Appendix

Using ETU with Autokey Batch Files

    Modifying an Autokey Batch File - Example

## About This Appendix

ETU 5.01 supports all the command functions of earlier versions of ETU. However, the menus for ETU 5.01 have been enhanced. This may cause some Autokey batch files that were used with earlier versions of ETU not to execute properly.

This appendix explains which types of Autokey batch files may need to be modified, as well as how to modify them.

## Using ETU 5.01 and above with Autokey Batch Files

The menu for ETU 5.01 and above changed from earlier versions. Some menu options have been moved from the Main menu to a Utility menu, and most all ETU commands have received new menu option numbers. (For example, the FROM3XB command in version 5.0 is option 13. In earlier versions of ETU, the FROM3XB command is option number 1.) These changes may cause some existing Autokey batch files not to work properly.

If you have an Autokey batch file that uses an ETU command line character string (like the one shown below), the batch file should work with ETU 5.01 just as it did with earlier versions.

user[FLX]
pgrmr[ENT]
to3xb qfname1('a:\sales.wed') tofile(wedsales) recl(132)ENT]
[ETU]

If you use an Autokey batch file that calls an ETU command from the ETU menu (like the example shown below), you must modify the batch file (due to changes in the ETU menu). An example of how to modify a batch file is shown below.

user[FLX]
pgrmr[FLX]
wsmenus[ENT]
16[ENT]
a:\sales.wed[FLX]
wedsales[FLX]
[ENT]
[ETU]

### Modifying an Autokey Batch File – Example

The Autokey batch file in the example above calls option 16 from the ETU menu. In versions of ETU prior to version 5.01, option 16 was the menu option for the TO3XB command. Because of changes to the ETU menu, the TO3XB command is now menu option 4.

To modify the Autokey batch file in the above example, change line 4 so that it reads 4[FLX] instead of 16[FLX]. Use any text editor or word processor to change the batch file. This change enables the Autokey batch file to be compatible with versions of ETU later than 5.0.

# CREATING DDS SPECIFICATIONS

About This Appendix

Creating DDS Specifications

## About This Appendix

This appendix explains how to define AS/400 files using Data Description Specifications (DDS specs). DDS specs define field-oriented files on the AS/400, which are files in either \*TAB, \*BASICS, or \*DIF format. Before you can transfer a field-oriented file, the file must be defined with DDS specs. DDS specs are not needed when transferring straight text files (\*TEXT format).

DDS specs are created using the Programming Development Manager (PDM) utility on the AS/400. In order to use PDM you must have authority to it. Your system administrator can verify whether you will be able to use PDM.

To create DDS 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 DDS specs with PDM by providing an example for you to follow.



## Creating DDS Specifications

In order to transfer field-oriented data, like that found in spreadsheets or databases, you must create a template that tells the AS/400 where each field begins and ends, and what type of data will be in each field. This is done with DDS specs or F&I specs. See Appendix D for a detailed explanation of creating F&I specs. The PDM utility on the AS/400 is used to create the DDS.

In this example you will see how to create DDS specs that allow a PC file to be transferred and properly formatted on the AS/400. The PC file is called SAMPLE and was created using a PC spreadsheet application. The PC spreadsheet file is represented in Figure C-1 shown below. This spreadsheet contains four fields. Table C-1, at the bottom of this page, describes how you will define each field.

Figure C-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 C-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 a DDS specs for a file, use the PDM (Programming Development Manager) utility on the AS/400. The steps below and on the following pages describe how to create DDS specs for the file shown on the previous page.

1. Log on to the AS/400.
2. Type STRPDM on the command line of the AS/400, and then press ENTER. The AS/400 PDM screen, shown in Figure C-2, appears. Select option 3 and press ENTER to work with members.

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

Figure C-2 The AS400 PDM screen

```
AS/400 Programming Development Manager (PDM)

Select one of the following:

    1. Work with libraries
    2. Work with objects
    3. Work with members

    9. Work with user-defined options

Selection or command
====>

F3=Exit      F4=Prompt    F9=Retrieve   F10=Command entry
F12=Cancel   F18=Change defaults

(C) COPYRIGHT IBM CORP. 1981, 1996.
```

3. The Select Members screen, shown in Figure C-3, appears. Type the required data in each prompt as explained in the following instructions, and then press ENTER.

Figure C-3 The Select Members screen

```
Specify Members to Work With

Type choices, press Enter.

File . . . . . QDDSSRC      Name, F4 for list
Library . . . . . *LIBL      *LIBL, *CURLIB, name

Member:
Name . . . . . *ALL          *ALL, name, *generic*
Type . . . . . *ALL          *ALL, type, *generic*, *BLANK

F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel
```

Type QDDSSRC in the File prompt.

In the Library prompt, type the name of the library in which you would like to store your DDS. If you are unsure, consult your system administrator. This example uses QGPL, the general purpose library.

In the Member Name prompt, type SAMPLE.

Press ENTER.

4. The Work With Members screen, shown in Figure C-4, appears. Press F6 to create the member.

Figure C-4 The Work With Members screen

Work with Members Using PDM		NLRISC			
File . . . . .	QDSSRC	Position to . . . . .			
Library . . . . .	QGPL				
Type options, press Enter.					
2=Edit	3=Copy	4=Delete	5=Display	6=Print	7=Rename
8=Display description	9=Save	13=Change text	14=Compile	15=Create module...	
Opt	Member	Type	Text		
	BASICTST	PF			
	QDSIGNON				
	SAMPLDDS	PF	sample DDS spec		
Parameters or command					Bottom
==>					
F3=Exit	F4=Prompt	F5=Refresh	F6=Create		
F9=Retrieve	F10=Command entry	F23=More options	F24=More keys		

5. The Start Source Entry Utility (SEU) screen, shown in Figure C-5, appears. Type SAMPLE in the Source member prompt, and then press ENTER.

Figure C-5 The Start Source Entry Utility screen

```
Start Source Entry Utility (STRSEU)

Type choices, press Enter.

Source file . . . . . > QDDSSRC      Name, *PRV
Library . . . . . > QGPL           Name, *LIBL, *CURLIB, *PRV
Source member . . . . . SAMPLE     Name, *PRV, *SELECT
Source type . . . . . *SAME        Name, *SAME, BAS, BASP...
Text 'description' . . . . . *BLANK

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

6. The Edit screen, shown in Figure C-6, appears. Press F4 for prompts to begin entering data.

Figure C-6 The SEU Edit screen

```
Columns . . . :  1  80                      Edit
QGPL/QDDSSRC
SEU==>
SAMPLE
FMT PF  ....A.....T.Name+++++Rlen++TDpB.....Functions+++++
***** Beginning of data *****
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
***** End of data *****

F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F10=Cursor   F11=Toggle
F16=Repeat find   F17=Repeat change   F24=More keys

Member SAMPLE added to file QGPL/QDDSSRC.           +
```

7. To create the DDS, enter the letter R in the Name Type field and RECORD in the Name prompt as shown in Figure C-7, and then press ENTER. This identifies this record as the record format name for the physical file defined by these DDS specs.

For each prompt, you can press HELP to receive more information.

Figure C-7 The SEU Edit Screen after F4 prompt

```

Columns . . . :   1 80                               Edit
QGPL/QDDSSRC
SEU==>
SAMPLE
  FMT PF .....A.....T.Name+++++Rlen++TDpB.....Functions+++++
***** Beginning of data *****
0001.00
***** End of data *****

                                Prompt type . . . PF      Sequence number . . . 0001.00

                                Name                                Data      Decimal
                                Type      Name      Ref      Length  Type      Positions  Use
                                R      RECORD
                                Functions

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

```

8. The DDS Edit screen, shown in Figure C-8, appears. Position the cursor in column 1 of line 0001.00, then type the letters IP. Press ENTER.

The letter I places SEU in insert mode and the letter P allows you to be prompted to enter the next record.

Figure C-8 The SEU Edit Screen

```
Columns . . . :   1 80                               Edit
QGPL/QDDSSRC
SEU==>
SAMPLE
FMT PF .....A.....T.Name+++++Rlen++TDpB.....Functions+++++
***** Beginning of data *****
IP01.00                R RECORD
030107
***** End of data *****

F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F10=Cursor  F11=Toggle
F16=Repeat find  F17=Repeat change  F24=More keys
```



9. The 0001.00 edit screen, shown in Figure C-9, appears. Type the required data in each prompt as explained in the following instructions, and then press ENTER.

Figure C-9 SEU Edit screen

```

Columns . . . :   1 80                               Edit
QGPL/QDDSSRC
SEU==>
SAMPLE
  FMT PF .....A.....T.Name+++++Rlen++TDpB.....Functions+++++
***** Beginning of data *****
0001.00          R RECORD
***** End of data *****

Prompt type . . . PF      Sequence number . . . ' ' ' ' ' ' ' '

Name                                     Data      Decimal
Type      Name      Ref      Length  Type      Positions  Use
          NAME                                     17      A

Functions

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

```

Type NAME in the Name prompt. (The NAME field is the first field defined in this file, shown in Figure C-1, followed by the fields TITLE, SALARY, and CITY.)

Type the number 17 in the Length prompt (be sure to press FIELD EXIT to move to the next field).

Type the letter A in the Data Type prompt.

Press ENTER.

10. The 0002.00 edit screen, shown in Figure C-10, appears. Type the required data in each prompt as explained in the following instructions, and then press ENTER.

Figure C-10 The SEU Edit screen

```

Columns . . . :   1  80                               Edit
QGPL/QDDSSRC
SEU==>
SAMPLE
  FMT PF .....A.....T.Name+++++Rlen++TDpB.....Functions+++++
***** Beginning of data *****
0001.00          R RECORD
0002.00          A          NAME          17A
***** End of data *****

Prompt type . . . PF      Sequence number . . . ' ' ' ' ' '

Name
Type      Name      Ref      Length      Type      Decimal
              TITLE              35      A
Functions

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

```

Type TITLE in the Name prompt.

Type the number 35 in the Length prompt (be sure to press FIELD EXIT to move to the next prompt.)

Type an A in the Data Type prompt.

Press ENTER.

11. The 0003.00 edit screen, shown in Figure C-11, appears. Type the required data in each prompt as explained in the following instructions, and then press ENTER.

Figure C-11 The SEU Edit screen

```

Columns . . . :   1 80                               Edit
QGPL/QDDSSRC
SEU==>
SAMPLE
  FMT PF .....A.....T.Name+++++RLen++TDpB.....Functions+++++
***** Beginning of data *****
0001.00          R RECORD
0002.00          A          NAME          17A
0003.00          A          TITLE         35A

***** End of data *****

Prompt type . . . PF      Sequence number . . . '*****'

Name                                     Data      Decimal
Type      Name      Ref      Length  Type      Positions  Use
          SALARY                                8        S          2

Functions

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

```

Type SALARY in the Name prompt.

Type the number 8 in the Length prompt (be sure to press FIELD EXIT to move to the next field.)

Type an S in the Data Type prompt.

Type the number 2 in the Decimal Positions prompt.

Press ENTER.

12. The 0004.00 edit screen, shown in Figure C-12, appears. Type the required data in each prompt as explained in the following instructions, and then press ENTER.

Figure C-12 The SEU Edit screen

```

Columns . . . :   1 80                               Edit
QGPL/QDDSSRC
SEU==>
SAMPLE
  FMT PF .....A.....T.Name+++++Rlen++TDpB.....Functions+++++
***** Beginning of data *****
0001.00          R RECORD
0002.00          A          NAME          17A
0003.00          A          TITLE         35A
0004.00          A          SALARY        8S 2
***** End of data *****

Prompt type . . . PF      Sequence number . . . ' ' ' ' ' '

Name                                     Data      Decimal
Type      Name      Ref      Length  Type      Positions  Use
          CITY                                     16      A

Functions

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

```

Type CITY in the Name prompt.

Type the number 16 in the Length prompt (be sure to press FIELD EXIT to move to the next field.)

Type an A in the Data Type prompt.

Press ENTER.

13. The Edit screen, shown in Figure C-13, appears. When you have defined each field, press F3 to exit the SEU.

The DDS for the example is now complete. Press F3 to exit.

If you made a mistake, place the cursor on the line requiring the change, and then press F4. To insert a line, place the cursor on the line that you want the new line to appear after, and then press F4. Otherwise, press F3 to exit.

Figure C-13 The SEU Edit screen

```

Columns . . . :   1  80                               Edit
QGPL/QDDSSRC
SEU==>
SAMPLE
  FMT PF .....A.....T.Name+++++Rlen++TDpB.....Functions+++++
***** Beginning of data *****
0001.00          R RECORD
0002.00          A          NAME          17A
0003.00          A          TITLE         35A
0004.00          A          SALARY        8S 2
0005.00          A          CITY          16A
***** End of data *****

                                  Prompt type . . .   PF      Sequence number . . .   '*****'

                                  Name                               Data      Decimal
                                  Type      Name      Ref      Length  Type      Positions  Use

                                  Functions

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

```

14. The Exit screen, shown in Figure C-14, appears. Enter a text comment, if desired, and then press ENTER.

A text comment can help you remember the purpose of the DDS spec you create.

Figure C-14 The SEU Exit screen

```
Exit

Type choices, press Enter.

Change/create member . . . . . Y           Y=Yes, N=No
Member . . . . . SAMPLE           Name, F4 for list
File . . . . . QDDSSRC           Name, F4 for list
Library . . . . . QGPL           Name
Text . . . . . DDS Spec for sample file transfer

Resequence member . . . . . Y           Y=Yes, N=No
Start . . . . . 0001.00           0000.01-9999.99
Increment . . . . . 01.00           00.01-99.99

Print member . . . . . N           Y=Yes, N=No

Return to editing . . . . . N           Y=Yes, N=No

Go to member list . . . . . N           Y=Yes, N=No

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel
```

15. The PDM screen, shown in Figure C-15, appears. Press F3 to exit to the AS/400 main menu.

The demonstration file is now ready to be transferred to the AS/400. Be sure to verify that the file has been created by reading the member added to file message at the bottom of the screen.

Figure C-15 The PDM screen

```
Work with Members Using PDM                NLRISC

File . . . . . QDDSSRC
Library . . . . . QGPL                Position to . . . . .

Type options, press Enter.
2=Edit      3=Copy  4=Delete 5=Display    6=Print    7=Rename
8=Display description  9=Save 13=Change text 14=Compile 15=Create module...

Opt  Member      Type      Text
    BASICST      PF
    QDSIGNON
    SAMPLDDS      PF      sample DDS spec
    SAMPLE      PF      DDS spec for sample file transfer

                                                    Bottom

Parameters or command
====>
F3=Exit      F4=Prompt      F5=Refresh      F6=Create
F9=Retrieve   F10=Command entry  F23=More options  F24=More keys
Member SAMPLE in file QGPL/QDDSSRC changed with 5 records.
```

# CREATING F&I SPECIFICATIONS

About This Appendix

Creating DDS Specifications



## About This Appendix

On the AS/400, files can be defined with Data Definition Specifications (DDS) or 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 AS/400, 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 AS/400. 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 data bases, you must create a template that tells the AS/400 where each field begins and ends, and what type of data will be in each field. This is done through the use of Format and Input (F&I) specifications or DDS specifications. See Appendix C for a detailed explanation of DDS 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 AS/400. 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 AS/400. 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 AS/400.
2. Enter **STRSEU** on the command line of the AS/400, and then press **F4**. The Start Source Entry Utility (STRSEU) 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 (STRSEU)

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 which 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

```

Columns . . . :   1 80                               Edit
QGPL/QRPGSRC
SEU==>
SAMPLE
FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8

***** Beginning of data *****
0001.00
***** End of data *****

Prompt type . . .  **      Sequence number . . . 0001.00

Data area
..... 1 ..... 2 ..... 3 ..... 4 ..... 5 ..... 6
      FSAMPLE IP F      128          DISK

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 **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....RlenLK1AI0vKlocEDevice+.....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....RlenLK1AI0vKlocEdevice+.....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....RlenLK1AI0vKlocEdevice+.....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....RlenLK1AI0vKlocEDevice+.....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 602SALARY

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 60SALARY
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 AS/400 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

Exit		
Type choices, press Enter.		
Change/create member . . . . .	Y	Y=Yes, N=No
Member . . . . .	SAMPLE	Name, F4 for list
File . . . . .	QRPGSRC	Name, F4 for list
Library . . . . .	QGPL	Name
Text . . . . .		
Resequence member . . . . .	Y	Y=Yes, N=No
Start . . . . .	0001.00	0000.01-9999.99
Increment . . . . .	01.00	00.01-99.99
Print member . . . . .	N	Y=Yes, N=No
Return to editing . . . . .	N	Y=Yes, N=No
Go to member list . . . . .	N	Y=Yes, N=No
F3=Exit F4=Prompt F5=Refresh F12=Cancel		

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

# ERROR MESSAGES: NATIVE MODE

## About This Appendix

This appendix lists the ETU error messages that are most likely to occur when you use ETU. The messages appear in order of their ETU message number, which precedes each message.

For each message, you will find the error message number, message text, and possible causes and **Solutions**.



## 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>**.

NOTE: If the error message does not show up here, it is not an ETU error message.

NOTE: For **CPF** messages that are not in this document follow the above procedure using **WRKMSGF QCPFMSG**.

## TRANSFER PROBLEMS:

### PROBLEM:

When we go into the option to transfer a spool file to a pc file and enter the outq, no spool files show up so we can select the spool files from a list.

### CAUSE:

You must be the owner of the spool file and it must be held in the queue.

### SOLUTION:

You must be the user to submit the print job to become the owner of the spool file.

Second, after you submit the print job, (usually you will use a dumbie queue, that is offline all the time), you go to **WSMENUS** and select your **3** option, for "**Host spool file to PC prt file**", it will bring you to where you put the printer's **Output queue** name. Once you fill that in, you usually have to page down to the bottom to get to near the last entry in the queue, which should be yours.

Sometimes the last entry is a QPRTSPLQ HOLD file, so it is the next to last, unless you delay in accessing it and other jobs come along. Then take **2** on the file **2=Copyprt & Transfer**.

### PROBLEM:

File xxxxxx was not found on the PC disk/diskette.

### CAUSE:

The qualified file name is not correct.

- The slashes are leaning the wrong way.
- A file that has security set on it can also cause this.
- If an application has the file open it can lock the file.

### SOLUTION:

Go to the folder on the driver where the actual file to be transferred exists. Right-click on it and select Properties to make certain it is not.

The file name and folder can be long, and the extension up to 3 characters long, and have spaces.

## USER MESSAGES:

### PROBLEM:

**"USR0020 Translation failed. Specified record length differs from existing file."**

### CAUSE:

The translation failed because the record length (RECL) you specified in the translation command does not match the record length of the file being translated.

### SOLUTION:

Verify that the record length you specified matches the actual record length of the file.

### PROBLEM:

Getting "**USR0022 xxxx bytes moved to/from PC disk/diskette.**" ... after moving to a new system. He used to monitor for a CPF.

### CAUSE:

Usually this is a good message. Sometimes users monitor for CPF messages in batch files. A newer version of ETU might not generate the same CPF failure.

### SOLUTION:

Monitor for USR0022.

### PROBLEM:

**"USR0031 \*PGM &1.&2 already exists G to delete and continue, C to cancel transfer request".**

### CAUSE:

The program you specified in the **PGMTO3XB** transfer in the library you specified already exists on the host.

### SOLUTION:

If you want to overwrite the old host program with the new one, press G. This will permanently delete the old program.

If you want to cancel the transfer request, press C.

**PROBLEM:**

**"USR0032 Error detected while materializing program into AS/400 object file does not contain valid AS/400 object data".**

**CAUSE:**

The program you tried to transfer to the host from the PC was not previously transferred to the PC from the host.

**SOLUTION:**

In order to transfer a program from the micro to the host with the PGMTO3XB command, the program must have originally been transferred to the micro from the host with the PGMFROM3XB command

**PROBLEM:**

**"USR1255 File &1 was not found on the PC disk/diskette."**

**CAUSE:**

The "OPEN FILE" command sent to the PC emulator failed because the file name specified was not found in the current directory on the PC disk/diskette.

**SOLUTION:**

Use the native command mode for ETU400. Correct the error and retry the command.

**PROBLEM:**

**"USR1256 ETU transfer failed"** sometimes tied with message: **"COD0006 AA Unsuccessful. Device error. Error 1331"**

**CAUSE:**

This could be a Windows file-sharing violation.

**SOLUTION:**

Close the file before transferring it.

**CAUSE:**

The Qualified File Name could be wrong.

**SOLUTION:**

Make certain that the C:\ is not C:/  
Make certain the extension is correct.

**PROBLEM:**

**"USR4001 The PC disk/diskette is full - Transfer ended."**

**CAUSE:**

The disk or diskette to which you are transferring became full during the transfer causing the transfer to end. When transferring a file to the micro with ETU, the size of the file being transferred may not be greater than the capacity of the disk or diskette receiving the file.

**SOLUTION:**

Specify a different drive with a greater capacity.

**PROBLEM:**

**"USR5255 Invalid PC file name or no space for a new directory entry."**

Invalid PC file name -- or -- No space for a new director entry.

**CAUSE:**

The file name specified for the PC file is not a valid file name. If a file with that name exists and is open by another application, it can cause this.

The file may already exist and have the Read-only attribute set in it's Properties. Right-click on the file and select Properties to check this.

File names must follow the naming conventions of the system you are using. See Chapter 3, 'using ETU' for more information about microcomputer file names.

**SOLUTION:**

If a file with that name exists and is open by another application, close it. Retry the transfer using a new file name.

**CAUSE:**

The disk or diskette you specified is full and the transfer operation did not begin. ETU will not initiate a transfer operation to a full disk or diskette.

**SOLUTION:**

Replace the diskette or specify a different disk drive.

**CAUSE:**

Virus protection software prevents the writing.

**SOLUTION:**

Disable the Virus protection software or the feature that prevents disk writing.

**PROBLEM**

"**USR5255 Invalid PC file name or no space for a new directory entry**" occurs when doing back to back file transfers.

**CAUSE:**

Timing problem.

**SOLUTION:**

Put a 5-second delay between transfers.

**PROBLEM**

"**USR6255 Old PC file not found or new PC file name already exists for RENAME command.**"  
When he does TESTB, it indicates that the file exists, when he does a rename, it complains that it does not exist.

**CAUSE:**

The name you specified for the old PC file in the RENAMEB command was not found.

**SOLUTION:**

Verify the name of the file you want to rename and retry the operation.

**CAUSE:**

The name you specified for the new PC file already exists.

**SOLUTION:**

Enter a different name for the new PC file.  
Do not put the path in the '**New file name**' parameter of the RENAMEB command.

**PROBLEM**

"**USR7255 File not found on PC disk/diskette in the current directory.**"

**CAUSE:**

The PC file you specified was not found on the current PC disk or diskette. For micro to host file transfers, the micro file and directory (path) names specified must match exactly with the actual file and directory names.

**SOLUTION:**

Change the directory name (path) and retry the file transfer.

**PROBLEM:**

**"USR7900 System Error - INCORRECT SYSTEM SERIAL NUMBER, CALL SYSTEM SUPPORT STAFF"**

or on ETU36

**"USER 7900 Option (3) - error in host registration system, contact Emerald Technology"**

**CAUSE:**

Have had ETU/36 for over a year. They are upgrading to an AS/400.

**SOLUTION:**

Call sales to transfer the license for \$100 and get a new registration for the new machine.

**CAUSE:**

ETU/36 will force a registration if you go into the OWNERID screen.

**SOLUTION:**

Uninstall and reinstall.

The next four messages may occur when using the T03XB or PWRT03XB command to transfer a micro file to a new host file.

**PROBLEM**

**"USR8255 For FILETYPE equal to \*DATA or \*SAVF, SRCSEX must be equal to \*NO".**

**CAUSE:**

The values specified for the FILETYPE and SRCSEX parameters are not compatible with each other.

**SOLUTION:**

If the value for the FILETYPE parameter is \*DATA or \*SAVF, the value for the SRCSEQ parameter must be \*NO.

**PROBLEM**

**"USR8256 For FILETYPE equal to \*SAVF, TRANSLATE must be equal to \*NO".**

**CAUSE:**

The values specified for the FILETYPE and TRANSLATE parameters are not compatible with each other

**SOLUTION:**

If the value for the FILETYPE parameter is \*SAVF, the value for the TRANSLATE parameter must be \*NO. USR8257 For FILETYPE equal to \*SAVF, RECL must be equal to O or 528.

**CAUSE:**

The values specified for the FILETYPE and RECL parameters are not compatible with each other.

**SOLUTION:**

If the value for the FILETYPE parameter is ~SAVF, the value for the RECL parameter must be O or 528.



**PROBLEM**

**"USR8258 File &1 not found, RECL must be non-zero to create the file".**

**CAUSE:**

The host file you specified could not be found.

**SOLUTION:**

Specify a new file name or specify a record length (RECL) greater than zero preferably = or greater than the file on the PC.

**PROBLEM**

**"USR8261 When parameter FRMMBR is \*ALL, parameter TRANSLATE cannot be \*DIF".**

**CAUSE:**

The values specified for the FILETYPE and RECL parameters are incompatible with each other. \*DIF translation is not supported when transferring \*ALL members from a Host data file.

**SOLUTION:**

The value for the FRMMBR parameter was specified as \*ALL and the value for the TRANSLATE parameter was \*DIF. If the TRANSLATE type is \*DIF, change the FRMMBR value to either \*FILE, \*FIRST, or \*LAST.

**PROBLEM**

**"USR9013 DOS error code 0C - General failure (Probable error : media error)."**

**CAUSE:**

The diskette/disk adapter detected an error while trying to work with the diskette/disk.

**SOLUTION:**

Make sure the diskette drive closing mechanism is fully latched. Replace the diskette if in question. The diskette may have been damaged during storage or formatted incorrectly. If the error remains, or happens on a hard disk, then run DOS diagnostic programs to determine if the diskette/disk unit is having hardware problems.

(See Also MAC ERRORS at the end of this doc.)

**PROBLEM**

**"USR9970 Invalid DIF header statements in the PC formatted file".**

**CAUSE:**

While reading the DIF header statements, the DATA header item was not encountered or was incorrectly specified.

**SOLUTION:**

Check that the DATA record is followed by a record containing 0,0 and another record containing "".

**PROBLEM**

**“USR9971 A DIF record found in PC file has record Length greater than allowed maximum of 258”.**

**CAUSE:**

The library containing the ETU programs is not in your library list.

**SOLUTION:**

Add the library containing the ETU programs to your library list.

**CAUSE:**

While translating a DIF, BASICS, or TAB formatted file into the host format, field over 258 bytes was encountered. DIF fields are terminated by an End-of-File (normally 0D0A - carriage return/line feed), and BASICS/TAB are separated by a specified delimiter. Over 258 characters were encountered while processing the data file with no intervening end of field designation.

**SOLUTION:**

Insure that the PC data file is in the format specified for the translation type on the command. Correct the data file or command.

**PROBLEM**

**“USR9972 Translation encountered unexpected characters in a field defined as numeric”.**

**CAUSE:**

The translation was converting data to a numeric field and encountered an unexpected character

**SOLUTION:**

Verify that the PC file format and the Translation type on the ETU command match. If the formats are correct, verify the field definitions of the host file match the PC file.

**PROBLEM**

**“USR9985 Default data-area not found”.**

**CAUSE:**

The library containing the ETU programs is not in your library list.

**SOLUTION:**

Add the library containing the ETU programs to your library list.

**ADDLIBLE PCTRAN**

## PROBLEM

"**USR9986 File transfer was interrupted: ("r" to retry or "c" to cancel)**".

### CAUSE:

Only known cause was a remote controller.

### SOLUTION:

Examine the configuration options for the remote controller. Contact that vendor. Possibly reroute or replace controller.

## PROBLEM

When I try to do a transfer, I get a message after the EMULATOR CHECK that says "**USR9987 Device is not P.C. or emulator loaded incorrectly.**"

### CAUSE:

An ETU function has failed because of incorrect data received from the display screen. A device other than a PC responding to an ETU command can cause this error. A PC with an emulator program that does not support ETU can cause it. This error can also occur when using a supported emulator and the operator presses the "Enter" key (rather than "Hot Keying" to DOS and loading the compatibility program) when the Copyright screen is displayed.

ETU is currently only supported with NLynx/Decision Data's **ES32** emulation. There once was a DOS Compatibility disk with a program called IBMTRAN that enabled any emulation to work. This software only works with other DOS software using serial COM ports. It is not supported today because the Windows Registry makes this too difficult.

### SOLUTION:

If you have our ES32 emulation, make certain that ETU is enabled.

- Select the **Session** menu, then **Properties**.
- Select the **API** tab.
- Make certain there is a check in **Enable ETU**.

## PROBLEM

"**USR9988 ETU encountered information not identified as part of an ETU process.**" This is sort of a catchall for ETU problems not previously identified and clarified.

### CAUSE:

Using Reflection software. IBMTRAN requires 5250 emulation software. Reflection software does VT100 emulation.

### SOLUTION:

The only emulation supported today is ES32. See <http://www.nlynx.com/html/products.htm>. Verify that the device is a PC with an ETU compatible emulation product running, and then retry the request.

**PROBLEM**

"USR9988 ETU encountered information not identified as part of an ETU process." This is sort of a catchall for ETU problems not previously identified and clarified.

**CAUSE:**

Trying to type during a transfer.

**SOLUTION:**

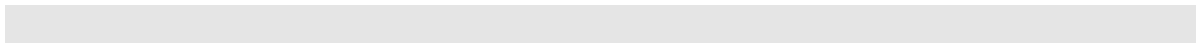
Do not key inside the emulation screen during a transfer.  
Do not press <Enter> on the message regarding the number of day's left.

**CAUSE:**

This could happen if the disk just got full.

**SOLUTION:**

Check the disk space on the PC.



## CPF MESSAGES:

These are the same as User Failures, except it was call by a program. Call Program Fault.

### PROBLEM

Getting “**CPF4101 File &2 in library &3 not found or inline data file missing**” when doing a file transfer.

### CAUSE:

The file was not opened. The reason code is below. The reason codes and their meanings are as follows:

- 01 - The library does not exist.
- 02 - The file does not exist. The library does exist.
- 03 - The file does not exist. The library specified as \*LIBL.
- 04 - The file was saved with storage freed.
- 07 - An inline data file does not exist.
- 08 - A diskette file with SPOOL (\*YES) is being opened for a read operation and an inline data file does not exist.
- 10 - The file on the remote system does not exist.
- 11 - The library on the remote system does not exist.

### SOLUTION:

When you get this error, move your cursor on top of the message and hit F1 to get more details.

Do one of the following based on the reason code shown, and try the request again:

- 01 - Change the library name with an appropriate override files command, or create a library (CRTLIB command) and a file with the appropriate name.
- 02 - Change the file name, the library name, or both the file name and library name with an appropriate override file command.
- 03 - Add the library to the library list (ADDLIBLE command) or change the file name, the library name, or both the file name and library name with an appropriate override file command.
- 04 - Restore the file (RSTOBJ command) or change the file name, the library name, or both the file name and library name with an appropriate override file command.
- 07 - Add an inline data file (DATA command) or change the file name, the library name, or both the file name and library name with an appropriate override file command.
- 08 - Specify the \*NO value for the SPOOL parameter on the Override Diskette File (OVRDKTF) command or add an inline data file (DATA command) or change the file name, the library name, or both the file name and library name with an appropriate override file command.
- 10 - Change the file name in the distributed data management (DDM) file with a Change DDM File (CHGDDMF) command or use the necessary procedures to make the file exist on the remote system.
- 11 - Change the library name in the distributed data management (DDM) file with a Change

DDM File (CHGDDMF) command or use the necessary procedures to make the library exist on the remote system.

**PROBLEM:**

Doing TO3XB on a field-oriented file. There is an error message "**CPF5029 Data mapping error on member &4**".

**CAUSE:**

A data mapping error occurred on member &4 file &2 in library &3, because the data fields in record number &6, record format &7, member number &8 are in error.

**SOLUTION:**

If the record number is zero, one of the following is at fault:

- The error occurred on an attempt to put or update a record through a logical file.
- The error occurred on a group by operation.
- The error was caused by a default value.
- The error was caused by a null value in the record.

See previously listed message **CPF5035** to determine which fields are in error. Change the fields that are in error and try the operation again.

If the record number is not zero, the error occurred on an attempt to read an existing record from a file. See the previously listed message CPF5035 to determine which physical file fields is in error. Then either delete the record from the file or change the data in the fields that are in error. Then try your request again.

**CAUSE:**

The AS400 date format is Century format, which has 3 digits for the year, 2 for the month and 2 for the day. The PC is usually month (2), day (12), and year 2001. This would confuse the AS400 and ETU; therefore ETU does not understand the data type of "L" for data type DATE.

**SOLUTION:**

Change this to a standard zoned decimal number field. You may need to change the dates in Excel to Century format, and then transfer to the AS400 using P (packed) instead of L.

**PROBLEM:**

Adv36 running SSP and OS400 environment. OS400 side running. **CPF5149 IO error was detected Xlation**. ETU400 - is launched by WSMENUS. File is located on the QS36F library.

**CAUSE:**

QS36F is not in the library list.

**SOLUTION:**

Add the QS36F library to the library list.

## PROBLEM

**CPF5149: I/O error detected in XLATIN.** Using 36 commands. Customer has to end the procedure that creates the file to be transferred before it will transfer successfully.

### CAUSE:

Need detailed printing of joblog using dspjoblog command. Prompt with F4 and change output to \*print to get details. Look for sharing errors or errors with subfile. Problem typically occurs in S36 mode. This is not an ETU problem but a system problem.

### SOLUTION:

Make certain that the system defaults for the **CRTPF** have **not** been modified **\*NO** for sharing.

## PROBLEM

**“CPF5035 Data mapping error on member XX”.**

### CAUSE:

A data mapping error occurred on field &4 in record number &5, record format &7, member number &8, in member &1 file &2 in library &3, because of error code &6. The error codes and their meanings follow:

- 1 -- There is data in a decimal field that is not valid.
- 2 -- A significant digit was truncated.
- 3 -- A floating point value exceeded the maximum representable value.
- 4 -- A floating-point value was less than the minimum representable value.
- 5 -- A floating point value that is not a number could not be mapped from double to single precision or from single to double precision.
- 6 -- A floating point value could not be mapped to packed decimal, zoned decimal, or binary.
- 8 -- A floating-point value that is not a number is not allowed as a key value.
- 9 -- Division by zero.
- 10 -- A partial binary key field could not be mapped. The length of the key did not include the entire binary field.
- 11 -- A partial key field could not be mapped. The length of the key did not include the entire binary, floating point, date, or timestamp field.
- 12 -- There is data in a DBCS field that is not valid.
- 13 -- Group by results could not be determined because no records were selected.
- 14 -- The length of a variable length field is not valid or the data in a substring variable is not valid.
- 15 -- A date value is greater than the maximum allowed value.
- 16 -- A date value is less than the minimum allowed value.

17 -- The format of the data in a date, time, or timestamp field is not valid.

18 -- There is data in a date, time, or timestamp field that is not valid.

19 -- An unexpected null field was found.

20 -- A field that is not 'null capable' could not be set to null.

21 -- The data has been converted from one CCSID to another. However, substitution characters were used for characters that could not be converted.

22 -- A shift-out character was found while converting to a single byte CCSID or a substitution character was found. The data could not be converted.

23 -- The data could not be converted from one CCSID to another.

24 -- The data could not be converted from an ASCII CCSID to an EBCDIC CCSID, or could not be converted from or to a UCS-2 (Universal Coded Character Set) CCSID. The addition of shift-out and shift-in characters or the expansion of single-byte data to a double-byte value caused the converted length to be larger than the maximum length the result could hold. If the error occurred on an attempt to read an existing record, file &2 in library &3 identifies the name of the physical file containing the field contributing to the mapping error. Otherwise, the error occurred on an attempt to put or update a record in a file and the file name identifies the open file that contains the field contributing to the mapping error. If the field name is \*N, the field name is not known or is a default value.

25 -- Hash value cannot be computed for the requested query.



## SYS MESSAGES:

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

### PROBLEM

He is 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 can not 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 *libname***
- 3) You may need to remove library list entries that are incorrect. **RMVLIBLE *libname***
- 4) If Use the newer commands of **MENU ETU36** and try it the manual first. The command line second, and OCL's third.
- 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**

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.

## OCL PROBLEMS

### PROBLEM:

When I submit my OCL, I get this message: "PCT0002 Invalid or missing parameter."

### CAUSE:

There are two steps to a transfer: the translation and the transfer itself. In OCL the parameters must be entered for each step.

### SOLUTION:

Use two steps: 1) translate, and 2) then transfer using \*NONE.

### PROBLEM:

The company has 8 output queues that will have new data every day for the same output queue names. They want to automate as much as possible the daily transfer process.

### ANSWER:

No matter what happens, the spool queue must be copied from the spool to a physical file. The ETU part is the translation from EBCDIC to ASCII and to move the file to the PC. ETU can only work with a physical file. The process of copying the spool file to a physical file use an OS/400 command called CPYSPLF. The command interface only allows for a single spool file to be copied. In general, the OS/400 spool file interface is very restrictive.

Bottom line, it does not matter what ETU command you want to use, you still have to copy the spool file to a physical file first. That is the part that is restrictive. There are ways around it but none of them are very generalized and would require programming.

Because the spool transfer functions do not generate a single command line, this is a function that would probably require two steps.

You would first use Copy prt spool file - PC/XLT: (PRTXTOPC) to perform an EBCDIC to ASCII translation. On a command line, type **PRTXTOPC** and press **F4**. Use **\*TRANSLATE** as the Action to perform. After successfully defining the parameters and submitting the command, use **F9** to retrieve the command for the batch file transfer.

Next use **PWRFROM3XB** to perform the actual transfer.

**PROBLEM:**

I am using ES32 macro commands to do transfers and I get this error message: "**USR9988 "ETU encountered information not identified as part of the ETU process"**".

**CAUSE:**

If a transfer has started, and more data comes in, this message will occur.

**SOLUTION:**

Put a delay between file calls, to make sure ETU finishes it's processes before another file is called.

**PROBLEM:**

How can I change the ETU utility to copy items to my C: drive instead of my A: drive?

**CAUSE:**

You must doing your ETU file transfers using a CL script. Otherwise when the menu selection for "Qualified file name": comes up, you would see that this is where to change from A:\ to C:\.

**SOLUTION:**

You will need to find the CL script and modify it. The line that needs to change will look similar to this: FROM3XB FRMFILE(GEORGE) QFNAME1('a:\george.txt') and you must change it to ('C:\george.txt')

**PROBLEM:**

When I try to do a transfer, I get a message after the EMULATOR CHECK that says, "**USR3255 return from native code processor"**".

**CAUSE:**

This is often the same error as USR9987, but occurs when command is called from a CL script. Technically it means "Select/Omit statement has too many parameters".

**SOLUTION:**

If you have our ES32 emulation, make certain that ETU is enabled.

- Select the **Session** menu, then **Properties**.
- Select the **API** tab.
- Make certain there is a check in **Enable ETU**.
- Also make certain that the CL script is called by the PC that is running our emulation.

**PROBLEM:**

**"CPF0818 Value cannot be converted to type implied by receiver".**

**CAUSE:**

The value to be converted and the receiving field are different data types.

**SOLUTION:**

Change the contents of the field to be converted or the type of the receiver field, and then try the

Change Variable (CHGVAR) command again. More information on data conversion in control language (CL) programs and proper formats of different constant types can be found in the CL Reference manual.

**CAUSE:**

Made a change to the "system message reply" so that they didn't transfer empty files and this is when they started to get the error.

**SOLUTION:**

Change it back and IPL.

**PROBLEM:**

I can run ETU from the menu works fine, but it fails when I run it from a procedure. **"CPF5149 I/O error was detected in XLATN - DOS 4.10 Emulation"**. Message is displayed when it starts to do the translating. It does not get this message when executed through the menu option 13.

**CAUSE:**

A parameter in the line has been entered incorrectly.

**SOLUTION:**

Try adjusting the spaces, look carefully at 1's and l's; 0's and O's.

**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 his 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.

## 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 can not start the OCL script from a batch file; it must be started interactively.

## 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 can do a transfer, but when I try to do the same command in a batch procedure, it fails. When ETU is beginning the transfer... it generates these error messages:

ETU transfer failed (see Job Log and ETU Audit Report) since **all FROM or TO devices failed**.

Devices in error are xxxxxxxx.

### CAUSE:

The device name or REQUESTER that you used for the “**Xfer from which micro?**” parameter is incorrect. Device not found during acquire of program device xxxxxx to file PWRWS library ETUNEW.

### SOLUTION:

The name of your device can be seen under the Menu – WSMENUS in the upper left. If you use **F9** to replay the command the parameter to fix is **MICLST**.

The device that you are going to use for the \*REQUESTOR must be connected and at the signon screen.

**PROBLEM:**

ETU requires that a session be at the signon screen and not logged on in order to accept batch transfers to or from a \*REQUESTOR.

Users have a tendency log on any session that they see.

**SOLUTION:**

This solution, for Windows 2000 only, allows you to configure a session that will run in the background.

As the Windows 2000 PC Administrator, use the Task Scheduler create a task using the browser to point to the first TVD. You will select the 'When I log on' option, and put the user name and password of the person who will use the PC.

When that person logs on, the task scheduler starts the TVD, but does not bring a screen up. You can only see EMUAPP32 in the task scheduler status as running or by looking at the tasks list.

The user can start another TVD by selecting a TVD shortcut. If you then look at the tasks list, the emuapp32 is running twice. For what it is worth, the first uses a lot more memory.

**PROBLEM:**

When running a CL procedure, you get these errors. Error RPG0202 caused program PWRFRM3X01 to stop. Function check. RPG9001 unmonitored by PWRFROM3XB at statement 118500, ERROR --- CPF9999 --- RETURNED FROM NATIVE CODE PROCESSOR. If you execute lines 1-9 in "SEU Procedure1", it crashes. If you execute lines 1-8 in "SEU Procedure1", then execute line 9 in "SEU Procedure2", it will work.

**CAUSE:**

CL procedure that was calling the programs to create the files was the same CL procedure that was calling the ETU commands to transfer/translate those same files. According to the CL documentation, the program that created the files has "ownership" of them, causing a SHARE violation when the ETU command tried to access them.

**SOLUTION:**

There is supposed to be a way to get around this using the SHARE command, however you can make two completely separate procedures; one to create the files and one to transfer the files.

# ASCII-EBCDIC PRINTABLE CHARACTERS



## About This Appendix

This appendix lists the ASCII and EBCDIC characters and their hexadecimal equivalents. Use this table when you create or modify an ETU translation table.

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		&
39	27		'
40	28		(
41	29		)
42	2A		*
43	2B		+
44	2C		,
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	3C		
61	3D		=
62	3E		
63	3F		?
64	40	(space)	@
65	41		A
66	42	â	B
67	43	ä	C
68	44	à	D
69	45	á	E
70	46		F
71	47	ã	G
72	48	ç	H
73	49	ñ	I
74	4A	ç	J
75	4B	.	K
76	4C	<	L
77	4D	(	M
78	4E	+	N
79	4F		O
80	50	&	P
81	51	é	Q
82	52	ê	R
83	53	ë	S
84	54	è	T
85	55	í	U
86	56	î	V
87	57	ï	W
88	58	ì	X
89	59	ß	Y
90	5A	!	Z
91	5B	\$	[
92	5C	*	\
93	5D	)	]
94	5E	;	^
95	5F	¬	~
96	60	-	`
97	61	/	A
98	62	â	B
99	63	Ä	C
100	64	à	d
101	65	á	e
102	66		f
103	67	À	g
104	68	Ç	h
105	69	Ñ	i
106	6A	!	j
107	6B	,	k
108	6C	%	l

109	6D	—	m
110	6E	>	n
111	6F	?	o
112	70	ø	p
113	71	É	q
114	72	è	r
115	73	ë	s
116	74	è	t
117	75	í	u
118	76	î	v
119	77	ï	w
120	78	>	x
121	79	`	y
122	7A	:	z
123	7B	#	{
124	7C	@	
125	7D	'	}
126	7E	=	~
127	7F	"	
128	80	ø	
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	j	
139	8B	k	
140	8C	l	
141	8D		
142	8E		
143	8F		
144	90		
145	91	j	
146	92	k	
147	93	l	
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		
158	9E		
159	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		
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	D5	N	
214	D6	O	
215	D7	P	
216	D8	Q	
217	D9	R	
218	DA		
219	DB		
220	DC		

221	DD		
222	DE		
223	DF		
224	E0		
225	E1		
226	E2	S	
227	E3	T	
228	E4	U	
229	E5	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		
241	F1		
242	F2		
243	F3		
244	F4		
245	F5		
246	F6		
247	F7		
248	F8		
249	F9		
250	FA		
251	FB		
252	FC		
253	FD		
254	FE		
255	FF		

# System/36-mode Commands

This appendix contains the commands used to emulate ETU36.

## About This Appendix

Some users of ETU/400 may be more familiar with the operating system of System/36 than with that of the AS/400. For this reason, ETU/400 supports a set of commands in System/36 mode, in addition to the native AS/400 commands.

The commands that are supported in both the native mode and System/36 mode are listed in this chapter. Any native command can be used by a System/36-mode OCL procedure. The syntax of the native command when used in OCL is the same as entering the command from the System/36-mode command line (all uppercase letters).

However, most native commands do not return status codes in the LDA so the abnormal completion of a native command cannot be detected by the OCL procedure. For this reason, System/36-specific commands are available in ETU/400.

This appendix describes how to access the System/36-mode menu and how to execute the commands. Finally, this appendix describes each command available in System/36 mode, listed in alphabetical order.



## Starting ETU in System/36 Mode

To use the Emulator Transfer Utility (ETU) in System/36-mode, follow these steps.

1. Start your emulation software on your micro.
2. Sign on to the AS/400 by entering your user I.D. and password
3. Start the System/36 mode on the AS/400. On the AS/400 command line, type:

**strs36**

or designate \*S36 on the user's profile in the special environment (SPCENV) parameter.

4. Access the ETU36 menu. On the AS/400 command line, type:

**menu etu36**

The ETU System/36-mode menu displays.

### The ETU36 Menu

The ETU36 menu supports most ETU native commands. 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.

Figure G-1 The ETU36 menu

```
Menu - ETU36                               Workstation ID - WH
      Emulator Transfer Utility S/36
      COPYRIGHT (c) 1998, Nlynx Systems

      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 menu option executes a ETU36 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 PRNTTOPC 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 ETU36 procedures are described alphabetically in this chapter starting on page G-6.

## Running ETU in System/36 Mode

Once you have signed on to the AS/400, you may execute an ETU procedure in the following ways:

- Prompt an ETU procedure by selecting a menu item from the ETU36 menu. To display the ETU 36 menu, type:

**MENU ETU36, your ETU library name**

- Prompt an ETU procedure by entering "ETU36 procedure", where procedure is the ETU procedure (such as ALLOCATE, FILETOPC, LIBRFRPC) to be executed. For example:

**etu36 allocate**

- Include in an OCL procedure.

## Native Commands Not Supported in the System/36 Mode

The System/36 mode of ETU/400 does not support the following ETU native-mode commands and utilities:

EDITTABLE, EDTUDF, PCFROMPC, PCTOPC, PWRFROM3XB, PWRT03XB, SBMBATXFER, SBMSCHJOB, STRSEU, VWSAVRQS.

## System/36-mode Commands and Native Mode Equivalents

The System/36-mode ETU commands and their equivalent commands in native mode are listed below in Table G-1.

Table G-1 ETU's System/36-mode commands and their native mode equivalents

<i>System/36-mode commands</i>	<i>Their native mode equivalents</i>
ALLOCATE	ALOCATB
DELETE	DELETEDB
COMPILE	STRSEU
FILEFRPC	TO3XB
FILETOPC	FROM3XB
LIBRFRPC	PGMTO3XB
LIBRTOPC	PGMFROM3XB
PRNTFRMPC	PRTTO3X
PRNTTOPC	PRTFROM3X
RENAME	RENAMEB
TESTFILE	TESTB
XLT36FIL	XLATETO
XLT36PRT	PRTTOPC
XLTPCFIL	XLATEFROM
XLTPCPRT	PRTXFRMPC

## ALLOCATE Procedure

```
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 \_\_\_\_.

## **COMPILE Procedures**

The translation table functions are not provided in the S/36 mode. The native-mode translate tables provide the required support. If you execute the COMPILE procedure, it calls the native-mode ETU command EDITTABLE (used to compile the translate source member).

See the EDITTABLE command in Chapter 10 for more information.

## DELETE Procedure

```
ETU DELETE workstation file name ... notify
                               NO
                               YES
```

The DELETE procedure deletes a file on the workstation.

**workstation file name** The name of the workstation file to be deleted. For additional information, see "Workstation File Name" on page \_\_\_\_.

**notify** Specifies whether you should be notified with a halt message if the workstation file is not found in the directory (optional). 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**

The translation table functions are not provided in the S/36 mode. The native-mode translate tables provide the required support. If you execute the EDITABLE function, it calls the native-mode ETU command STRSEU (used to edit the source member).

See the STRSEU command in Chapter 10 for more information.

## FILEFRPC Procedure

```
ETU36 FILEFRPC, workstation file name , AS400 label, record len1 , T,  
                                128
```

```
reformat      xlate file  xlate size  record len2  
*TEXT        ATOE1      1000      record len1
```

```
truncate  
YES  
NO
```

```
ETU36 FILEFRPC, workstation file name , AS400 label, record len1 , T,  
                                128
```

```
reformat      xlate file  xlate size  input specs ,  
*DIF          1000
```

```
*BASICS  
*TAB
```

```
spec lib  
clilb
```

```
ETU36 FILEFRPC, workstation file name , AS400 label, record len1 , T,  
                                128
```

```
*SAVE  
*NONE
```

The FILEFRPC procedure transfers a file from the workstation to the AS/400, optionally translating the file into an AS/400 file format.

**workstation file name** The name of the workstation file to be transferred from the workstation. For additional information, see "Workstation File Name" on page \_\_\_\_\_.

**AS/400 label** The file label (name) of the AS/400 file to receive the file transferred from the workstation.



**record len1** (optional) is the record length of the target AS/400 file ("AS/400 label"). This value defaults to 128.

**retain** (optional) is the file retention (how long the file is to exist) for the translated "AS/400 label2". 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 AS/400 job. The default value is T.

**reformat** (optional) specifies the type of data translation to take place on the AS/400 file.

**\*TEXT (or \*YES)** the file is to be translated into ASCII text format.

**\*DIF** the file is to be translated into DIF format.

**\*BASICS** the file is to be translated into BASIC Sequential format.

**\*TAB** the file is to be translated into BASIC Sequential format with tab characters as the field delimiters.

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

**\*NONE** **no** translation of the file is to be performed (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 EBCDIC, 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. Translation file members ATOE1 and ATOE3 are provided on the AS/400 distribution diskette and are used to translate from the ASCII to the EBCDIC character set. See the EDITABLE and COMPILE procedures for more information on creating new translation table files. The default is ATOE1.

<p><b>Mac Users:</b> Do <b>not</b> use the "xlate file" default value. You must use the value ATOE1M.</p>
---

**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 len1").

**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 "AS/400 label" is to be omitted. Specifying NO causes new records 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

```
ETU36 FILETOPC, AS/400 label, date ,  
  
      reformat , xlate file , xlate size , record len  
      *TEXT      ETOA1          1000          128  
  
      , workstation file name
```

```
ETU36 FILETOPC, AS/400 label, date ,  
  
      *DIF      , xlate table , xlate size , input specs  
      *BASICS          1000  
      *TAB  
      spec lib , workstation file name  
      clib
```

```
ETU36 FILETOPC, AS/400 label, date , *SAVE  
      *NONE  
  
      , workstation file name
```

The FILETOPC procedure transfers an AS/400 file from the AS/400 to the workstation specified, and optionally translates the file into a workstation data format.

- AS/400 label** is the label (name) of the file to be sent to the workstation.
- date** (optional) is the date of file "AS/400 label" (YMD format).
- Reformat** (optional) specifies the type of data translation to take place on the AS/400 file.
- \*TEXT (or \*YES) the file is to be translated into an ASCII text format.
  - \*DIF the file is to be translated into a DIF format.
  - \*BASICS the file is to be translated into a BASIC Sequential format.
  - \*TAB the 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 no translation of the file is to be performed (the default).

When using \*SAVE to move executable programs between workstations, all necessary data (including Mac data and resource forks) is transferred. Data must have been previously transferred with the FILEFRPC procedure with reformat type \*SAVE.

**xlat file** (optional) is the translation table file. Translation file member ETOA1 is provided on the AS/400 distribution diskette and can be used to translate from the EBCDIC to the ASCII character set in variable length records. Translation file member ETOA2 is provided on the AS/400 distribution diskette and can be used to translate from the EDCDIC to the ASCII character set in fixed length records. See the EDITABLE and COMPILE procedures for more information on creating new translation table files. The default value is ETOA1.

<p><b>Mac Users:</b> Do <b>not</b> use the default "xlat file" value. You must use either ETOA1M for variable-length records or ETOA2M for fixed-length records.</p>
--

**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 translated (workstation-formatted) AS/400 file to be sent to the workstation. This parameter is ignored if translation is not performed. The default value is 128.

**input specs** is the name of the field definition member in library "spec lib" that defines the fields of the file being translated. 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 F and I RPG specifications.

**spec lib** (optional) is the library containing the field definition member ("input specs"), and the translation 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 AS/400. For additional information, see "Workstation File Name" on page 4-2.

## LIBRFRPC Procedure

```
ETU36 LIBRFROMPC, workstation file name , record len ,
                                     128

      reformat , xlate file , xlate size , record len2
*TEXT      ETOA1      1000      record len1

      , truncate
      *YES
      *NO
-----

      *DIF      , xlate table , xlate size , input specs
*BASICS      1000
*TAB
      spec lib , workstation file name
      clib
-----

      *NONE

lib name      replace
clib      YES
           NO

      add ctl      member type      file size
YES, member name P      xlate size
NO      S
```

The LIBRFRPC procedure transfers and optionally translates a workstation file into library members on the AS/400.

**workstation file name** is the name of the workstation file to be transferred to the AS/400. For additional information, see "Workstation File Name" on page 4-2.

**record len1** (optional) is the record length of the library members when they are placed in the library. The default value is 120.

**reformat** (optional) specifies the type of data translation to take place on the AS/400 file.

\*TEXT (or \*YES) the file is to be translated into an ASCII text format.

\*DIF the file is to be translated into a DIF format

\*BASICS the file is to be translated into a BASIC Sequential format.

\*TAB the file is to be translated into a BASIC Sequential format with tab characters as the field delimiters.

\*NONE no translation of the file is to be performed (the default).

**xlat file** (optional) is the translation table file on the AS/400 disk. Translation file members ATOE1 and ATOE3 are provided on the AS/400 distribution diskette for translation from the ASCII to the EBCDIC character set. See the EDITABLE and COMPILE procedures for more information on creating new translation table files. The default value is ATOE1.

<b>Mac Users:</b>	Do <b>not</b> use the default "xlate file" value. You must use the value ATOE1M.
-------------------	--

**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 library member work file ("record len1").

**input specs** is the name of the field definition member in library "spec lib". These are standard F and I (file/input) RPG input specifications. See Chapter 5, "File Transfer Facility," for more information on creating F and I RPG specifications.

**truncate** (optional) is YES if data exceeding the record length of the library member work file is to be omitted. Specifying NO causes a new record to be written to contain the overflow. This parameter is ignored if translation is not performed. The 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.

**lib name** (optional) is the name of the library on the AS/400 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 workstation 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

```
ETU36 LIBRTOPC, member name, partall      member type      member lib
      member name                          PROC             clib
      partial name, ALL                     LIBRARY
      ALL,                                  SOURCE

      record len 1 ,   file size           remove ctl
                        1000                YES
                                                NO

      reformat ,   xlate file ,   xlate size , record len2
      *TEXT       ETOA1           1000       record len1

      *DIF      ,   xlate table ,   xlate size , input specs
      *BASICS   ,                               1000
      *TAB

      *none , , , ,

      workstation file name
```

The LIBRTOPC procedure transfers a AS/400 library member from the AS/400 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.

**partall** (optional) if the member name is only a partial name, enter ALL to have all members of the specified type extracted from the library. Leave this entry blank if the member name is complete.

**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.

The LOAD and SUBR member types are not supported in the S/36 environment. If you wish to transfer these member types from the host to a micro, use the native command PGMFRM3XB or the S/36 environment command FROMLIBR to copy library members to a save file and then use the native command PWRFROM3XB to transfer the save file (object type \*SAVF) from the host.

**member lib** (optional) is the name of the AS/400 library from which to extract the specified members. The default value is the current library.



**record len1** (optional) is the record length of the 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 AS/400 utility control (\$MAINT) 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 on the AS/400 file.

- \*TEXT (or \*YES) the file is to be translated into an ASCII text format.
- \*DIF the file is to be translated into a DIF format.
- \*BASICS the file is to be translated into a BASIC Sequential format.
- \*TAB the file is to be translated into a BASIC Sequential format with tab characters as the field delimiters.
- \*NONE **no** translation of the file is to be performed (the default).

**xlat file** (optional) is the translation table file. Translation file member ETOA1 is provided on the AS/400 distribution diskette and can be used to translate from the EBCDIC to the ASCII character set in variable length records. Translation file member ETOA2 is provided on the AS/400 distribution diskette and can be used to translate from the EBCDIC to the ASCII character set in fixed length records. See the EDITABLE and COMPILE procedures for more information on creating new translation table files. The default value is ETOA1.

**Mac Users:** Do **not** use the default value. You must use either ETOA1M for variable length records or ETOA2M for fixed length records.

**record len2** (optional) is the record length of the translated (workstation-formatted) AS/400 file to be sent to the workstation. This parameter defaults to "record len1", and is ignored if translation is not performed.

**input specs** is the name of the field definition member in library "spec lib" that defines the fields of the file being translated. These are standard F and I (file/input) RPG specifications, defining the file and its fields. See Chapter 4, "File Transfer Facility," for more information on creating F and I RPG specifications.

**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 library members from the AS/400. For additional information, see "Workstation File Name" on page 4-2.

## PRNTFRPC Procedure

```
ETU36 PRNTFRPC, workstation file name , , , #copies , printer id

                forms id   cpi   lpi   align
                , 10      10   , YES
                15      15   , NO

                hold       width   translate table
                YES ,     198 ,     ATOE3
                NO       132
```

The PRNTFRPC procedure transfers and translates a PC-DOS file and places it on the AS/400 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 to the AS/400. 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 AS/400. The default is one copy.

**printer id** (optional) is the AS/400 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 ATOE3.

## PRNTTOPC Procedure

```
ETU36 PRNTTOPC, function  spool name , xlat file , qualifier  file size
          COPYPRT ,      ALL          ETOA1          RELEASE    1000
                          Fxxx
                          Spoolid          CANCEL
          TRANSLATE, AS/400 label  xlat file  date  file size
                          ETOA1          1000
          from window ,  to window , workstation file name
          1              132
```

The PRNTTOPC procedure translates and transfers print spool reports from the AS/400 to the workstation.

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

- procedure** 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 AS/400 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 AS/400 print spool. Valid entries are the spool id (in the form SPxxxx), the forms type to be extracted (in the form Fxxxx), or ALL to extract all inactive print spool items. Examine the COPYPRT procedure in the manual System Reference for the AS/400 Environment for more information on these entries.
- AS/400 label** is the label (name) of the AS/400 "COPYPRT" print file existing on the AS/400 disk that is to be translated into the workstation print format.
- xlat file** (optional) is the translation table file. Translation file member ETOA1 is provided on the AS/400 distribution media and is used to translate from the ASCII to the EBCDIC character set. See the EDITABLE and COMPILE procedures for more information on creating new translation table files. The default is ETOA1.
- qualifier** (optional) is the action to be taken regarding the copied AS/400 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.
- date** (optional) is the date of the AS/400 COPYPRT file "AS/400 label" (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 that receives the print file from the AS/400. For additional information, see "About Microcomputer Files" on page 3-11.

## RENAME Function

```
ETU36 RENAME, workstation file name , , , new file name.
```

The RENAME procedure changes the filename of a workstation file.

**workstation file name** is the current name of the workstation file to be renamed. For additional information, see "Workstation File Name" on page 4-2. You may include the DOS pathname in the file specification (e.g., C:\TEST\MYFILE.TXT). 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. You cannot specify the drive or path. The new file name field is 80 characters to make it consistent with other workstation-file-name fields.

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

## TESTFILE Procedure

```
ETU36 TESTFILE, workstation file name , , , ida 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 tested.

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 1 to 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

```
ETU36 XLT36FIL, AS/400 label1, date , AS/400 label2, retain ,
                                     J
                                     T

reformat , xlate file , xlate size , record len ,
*TEXT    ETOA1      1000      128

                                     truncate
                                     YES
                                     NO

*DIF , xlat table , xlate size , input specs ,

*BASICS      1000
*TAB

                                     spec lib
                                     clib
```

The XLT36FIL procedure performs translation of a file residing on the AS/400 disk into a workstation data format. This procedure may be run from the job queue (with JOBQ), or released from the workstation (with EVOKE).

- AS/400 label1** is the label (name) of the AS/400 file to be translated.
- date** (optional) is the date of the AS/400 file to be translated (YMD format).
- AS/400 label2** 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 AS/400 disk when this procedure is called.
- retain** (optional) is the file retention (how long the file is to exist) for the translated "AS/400 label2". 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 AS/400 job. The default value is T.

**reformat** (optional) specifies the type of data translation to take place on the AS/400 file.

\*TEXT (or \*YES) the file is to be translated into an ASCII text format.

\*DIF the file is to be translated into a DIF format.

\*BASICS the file is to be translated into a BASIC Sequential format.

\*TAB the file is to be translated into a BASIC Sequential format with tab characters as the field delimiters.

\*NONE **no** translation of the file is to be performed(the default).

**xlat file** (optional) is the translation table file on the AS/400 disk. Translation file member ETOA1 is provided on the AS/400 distribution diskette and can be used to translate from the EBCDIC to the ASCII character set in variable length records. Translation file member ETOA2 is provided on the AS/400 distribution diskette and can be used to translate from the EBCDIC to the ASCII character set in fixed length records. See the EDITABLE and COMPILE procedures for more information on creating new translation table files. The default value is ETOA1.

**Mac Users:** Do **not** use the "xlate file" default value. You must use either ETOA1M for variable length records or ETOA2M for fixed length records.

**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 ("AS/400 label2"). 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. See Appendix D, "Creating F&I Specifications," 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

```
ETU36 XLT36PRT,  function  spool name ,  qualifier  ,
                  COPYPRT  ALL      RELEASE
                  Fxxx      CANCEL

                  TRANSLATE, AS/400 label1,  date1 ,
                  mode  , AS/400 label2,  file size  ,  retain,
                  CREATE      1000      J
                                      T
                  ADD,  AS/400 label2,  date2 , ,
                  from window  ,  to window
                  1      132
```

The XLT36PRT procedure translates AS/400 print spool items into an AS/400 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 AS/400 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 AS/400 print spool. Valid entries are the spool id (in the form SPxxxx), the forms type to be extracted (in the form Fxxxx), or ALL to extract all print reports. Examine the COPYPRT procedure in the AS/400 reference manual for more information on these entries.
- AS/400 label1** is the label (name) of the AS/400 COPYPRT print file existing on the AS/400 disk that is to be translated into the PC print format.
- qualifier** (optional) is the action to be taken regarding the copied AS/400 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 AS/400 COPYPRT file "AS/400 label1".

**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 "AS/400 label2". Specify ADD to add the print reports to the existing file "AS/400 label2". The default value is CREATE.

**AS/400 label2** 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 AS/400 file to be added to "AS/400 label2".

**retain** (optional) is the file retention (how long the file is to exist) for the translated "AS/400 label2". Allowed values are T or J. Specifying T will result 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 AS/400 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

```
ETU36 XLTPCFIL, AS/400 label1, date , AS/400 label2, retain,
                                     J
                                     T
reformat , xlate file , xlate size , record len,
*TEXT ATOE1 1000 128

date ,

truncate
YES
NO

*DIF , xlat file , xlate size , input specs,
*BASICS 1000
*TAB

spec lib
clib
```

The XLTPCFIL procedure translates a previously transferred workstation data file into an AS/400 formatted data file.

**AS/400 label1** is the name of workstation data file residing on the AS/400 that is to be reformatted for the AS/400.

**date** (optional) is the date of the "AS/400 label1" file (YMD format).

**AS/400 label2** is the label (name) of the file that will contain the results of the translation process. This name cannot already be on the AS/400 disk when this procedure is called.

**retain** (optional) is the file retention (how long the file is to exist) for the translated file "AS/400 label2". 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 AS/400 job. The default value is T.

**reformat** (optional) specifies the type of data translation to take place on the AS/400 file.

\*TEXT (or \*YES) the file is to be translated into an ASCII text format.

\*DIF the file is to be translated into a DIF format.

\*BASICS the file is to be translated into a BASIC Sequential format.

\*TAB the file is to be translated into a BASIC Sequential format with tab characters as the field delimiters.

\*NONE **no** translation of the file is to be performed (the default).

**xlat file** (optional) is the translation table file on the AS/400 disk. Translation file members ATOE1 and ATOE3 are provided on the AS/400 distribution diskette for translating from the ASCII to the EBCDIC character set. See the EDITABLE and COMPILE procedures for more information on creating new translation table files. The default value is ATOE1.

**Mac Users:** Do **not** use the default "xlat file" value. You must use the value ATOE1M.

**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 ("AS/400 label2"). 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 Appendix D for more information on creating F and I RPG specifications.

**truncate** (optional) is YES if data exceeding the record length of "AS/400label2" 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.

## XLTPCPRT Procedure

```
ETU36 XLTPCPRT,  AS/400 label,      date ,      # copies , printer id ,
                forms id    , cpi ,   lpi ,   align  hold
                        10   6   YES   YES
                        12   8   NO  NO
```

The XLTPCPRT procedure translates a PC-DOS file and places it on the AS/400 print spool.

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

- AS/400 label** is the label (name) of the previously transferred workstation print file on the AS/400 disk to be translated into AS/400 print report and placed on the AS/400 print spool.
- date** (optional) is the date of the workstation print file "AS/400 label" (YMD format).
- # copies** (optional) is the number of copies of the report to print on the AS/400. The default is one copy.
- printer id** (optional) is the AS/400 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.

# INTERNATIONAL TRANSLATION TABLES



## Installing International Translation Table

NLynx Technologies now supports sixteen languages for the ETU/400 file transfer programs on the AS/400 host computer:

<b>Country</b>	<b>ID</b>
Belgian French	BF
Canadian French	CF
Danish	DK
French	FR
German	GR
Italian	IT
Dutch	NL
Norwegian	NO
Portuguese	PO
Swiss French	SF
Swiss German	SG
Spanish	SP
Finnish	SU
Swedish	SV
U.K. English	UK
U.S. English	US (default)

If you are using a non-U.S. language on your AS/400, then you need to install the appropriate non-U.S. translation tables on your host machine. If you are using the U.S. English table, you do not need to install the files on the translation table diskette.

To install the translation tables on your AS/400, follow the steps below:

1. Insert the media on the host. Enter these commands:

```
RSTOBJ OBJ (SRCxx) SAVLIB (NEWTABLE) DEV (device-name) RSTLIB (yyyyyy)
```

where *xx* is the country ID, *device-name* is the name of the device on which you are installing the tables, and *yyyyyy* is the library containing ETU.

2. Add the library containing ETU to your library list, and enter this command:

```
CRRTABLE SRCxx TRANLIB (library-name)
```

where *xx* is the country ID and *library-name* is the library in which you want the new translation table to reside. You can place the translation table in any library including the library containing ETU.

# ETU Automation

## ETU400 Automation

This section is intended as an explanation of the considerations that one must make when one is trying to automate file transfers using ETU400. It is usually easier to handle the function of automating the process on the AS/400 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 ETU400 commands will work in interactive mode. Some ETU400 commands will work in batch mode. The ETU commands that work in batch mode are called power commands. Power commands are only ETU400 commands. If you run SSP, then you can't use ETU400, only ETU36. Technically you can use the power commands, but they are not supported in this environment.

### Interactive ETU Commands:

The following commands are interactive and can be run only from a microcomputer emulating a 5250 and using an ETU certified 5250 API: ALOCATB, DELETEDB, TO3XB, PGMTO3XB, PGMFROM3XB, RENAMEB, TESTB, FROM3XB, PRTFROM3X, PRPTO3X, EDITTABLE, and STRSEU.

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

Interactive commands can be put in a CL 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. See VWSAVRQS.

**NOTE:** You can not run the interactive commands from a CL 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.

### Power or Batch ETU Commands:

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

The ETU power commands can also be submitted to batch by using SBMJOB, or the ETU facility SBMBATXFER (submit batch transfer), or from the SBMSCHJOB (submit scheduled

job). The ETU power commands are PWRDELETEB, PCFROMPC, PCTOPC, PWRFROM3XB, PWRRENAMEB, and PWRTO3XB.

## AUTOMATION MECHANISMS

When you are setting up an automated application on the AS/400 there are two mechanisms with which to do this. There are CL scripts (called OCL scripts on the System 36) and batch files.

Batch commands are submitted to a job queue. The only batch commands that will work are the translation commands, and the power commands, as identified above. See SBMBATXFER, and SBMSCHJOB.

CL scripts are created using SEU (STRSEU). To the AS400, CL scripts look like interactive commands entered by the terminal that submits the CL 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. 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.

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.

**VWSAVRQS** will rerun commands that you saved by saying **\*YES** to the **Save request definition** question on the PCFROMPC, PCTOPC, PWRFROM3XB, PWRTO3XB, FROM3XB, or TO3XB procedure. (You may need to do a <Page Down> to see the 'Save' option.

When you submit one of the commands listed above, answer the '**Save request definition**' question with **\*YES**. When you want to resubmit the command, type **VWSAVRQS** on a command line to bring it up.

See page 9-17 and 10-56 of the ETU400 User Guide for specifics.

## SBMBATXFER

This command allows you to post a transfer from the AS/400 to a PC that has ES32 with ETU enabled, and is at the signon screen. This command works with PWRFROM3XB, PWRTO3XB, PCFROMPC, and PCTOPC. The PC that you are transferring to will be defined as the **\*REQUESTOR**.

See page 9-18 and 10-46 of the ETU400 User Guide for specifics.

## SBMSCHJOB

This command allows you to schedule a transfer from the AS/400 to a PC that will have ES32 with ETU enabled, and be at the signon screen, when the scheduled time occurs. The PC that you are transferring to will be defined as the **\*REQUESTOR**. This command works with any of the following

ETU commands: PWRFROM3XB, PWRT03XB, PCFROMPC, and PCTOPC, as well as non-ETU commands that can be submitted to a batch.

See page 9-21 and 10-47 of the ETU400 User Guide for specifics.

# ETU THROUGHPUT CONSIDERATIONS

## ETU THROUGHPUT CONSIDERATIONS

This is a guide to answer questions regarding throughput with ETU 400. The latest revision of ETU 400 is 5.02 for ETU 400 CISC and 5.04.1 for ETU 400 RISC. Use the ETUVER command to find what version you have.

### QUESTION:

#### What kind of throughput can I expect?

### ANSWER:

This would depend on several factors:

The connection to the host, twinax or TCP/IP is the first factor.

If the file requires translation from EBCDIC to ASCII it will add processing time.

If you are transferring a database file using a DDU or F & I specification it will take longer.

If you are running SSP it will slow it down.

Your AS400 CPU speed and OS/400 version will make a difference.

If there is a lot of other processing taking place on your host it will slow it down.

If the job is submitted to a batch queue, it will be slower.

If print jobs being sent to the same twinax port that the ES/PCI is on, it will slow things down.

If a network is congested, it could slow a transfer down.

Using TCP/IP, 750K per minute. Figure about 15 Megabytes per hour or 5000 cps.

For just a transfer, 30 pages, with 1953 records at 110 bytes each, for a total of 162694 bytes transfers in 20 seconds. This is about 8134 CPS.

### QUESTION:

What can I do to improve the throughput?

### ANSWER:

Move to a faster connection. Perform the transfer during times when the system and or network are not so busy.

Specify the exact form length and/or record length.

**QUESTION:**

What other options are there to ETU?

**ANSWER:**

FTP (such as File Transfer Protocol, using TCP/IP) or Client Access/400.

**QUESTION:**

What are the pros and cons of using FTP?

**ANSWER:**

FTP is fairly easy to use.

FTP does not support EBCDIC to ASCII, or Comma Separated Values, Basic Sequential, or Differential translations. CSV is a comma delimited file where fields in a record is delimited by a comma (commas in the "data" must be quoted).

FTP is supplied as part of the system on OS/400 and Windows 2000.

**QUESTION:**

What are the pros and cons of using ClientAccess/400?

**ANSWER:**

ClientAccess/400 is an option that you get with your AS/400 whether you want it or not. IBM calls this free. It can be difficult to install. Does not support batch capability, not invocable from the host (must be initiated from the client) and cannot transfer spool files. CA/400 does support CSV, Basic Sequential, or Differential translation. There are two file systems in the AS/400. The original QSYS file system where you have libraries, files and members. And there is IFS (Integrated File System) which is basically a similar file system to Unix (and DOS in some ways). Once installed and configured properly, the drag and drop capabilities are easy to use.

By the way, our ES/TCP has IFS support built in.

**QUESTION:**

What are the pros and cons of using ETU400?

**ANSWER:**

ETU is menu-driven and easy to use.

ETU supports CSV translation Basic Sequential, or Differential translation.

ETU can do batch transfers. See <http://www.nlynx.com/html/tb-etuauto.htm>

If the user wants to create "canned" transfers (something that the MIS guy builds and made available for all to use) or batch driven (run at a scheduled time), ETU is a good choice.

If you are looking for blazing speed, or time is a factor and the transferred file is enormous, then ETU is not the fastest choice. There are ways to improve it by using FTP.

**QUESTION:**

How would I use ETU400 and FTP?

**ANSWER:**

If you use ETU reasonably often and wish the file transfers quicker, for say files that are over 100,000 records, this is a way to do it.

1. Use ETU to only translate the file. Put the translated file in your private library. Don't use QTEMP. QTEMP exist only for the session that you are in. Does not work across sessions.

If the file on the AS/400 is not a spool file, take option 20 at the main ETU menu. For spool files, specify option 1 to select the file to be translated.

2. On your PC, open the DOS box (also called Command Prompt these days).

3. Type FTP to start FTP. Signon with your user id and password.

4. Type BINARY to stop FTP from translating the file. FTP can also translate the file but ETU have better translation features, such as comma delimited, etc.

If you use Query/400, it can output a physical file instead of spool file. You have to be careful with dates however. If Query outputs to a physical file, it does not reformat the data. This is an issue with dates because the AS/400 use the "century date format".

5. GET download the file, PUT upload the file. HELP will give you a summary of the FTP commands.

6. The only tricky bit is specifying the "file path" on the AS/400. It will in general look like:

```
QSYS.LIB/user_library.LIB/user_file.FILE/member.MBR
```

Note the "forward slash" and the qualifier after each object name (LIB is library, FILE is a physical file and MBR is a member in a physical file). Note also that you have to identify the member of the file. In general this defaults to the name of the file. For example, if I want to transfer file EXPENSE that exists in library QGPL, the command line will look like the following. It is assumed that the member name had defaulted to the file name.

```
GET QSYS.LIB/QGPL.LIB/EXPENSE.FILE/EXPENSE.MBR c:\temp\expense report.txt"
```

Note that the command interpreter expects file names (on the PC) with embedded blanks to be quoted.

7. The transfer is really quick.



# GLOSSARY

<b>Allocate</b>	To create a file on the microcomputer.
<b>API</b>	Application Program Interface. The interface (calling conventions) by which an application program accesses operating system and other services.
<b>ASCII character set</b>	American Standard Code for Information Interchange. A standard set of characters used by microcomputers to represent data. See also EBCDIC character set.
<b>BASIC sequential format</b>	*BASICS. 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.
<b>batch processing</b>	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.
<b>data transfer</b>	The process of moving data between the micro and the host with ETU.
<b>data translation</b>	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.
<b>DDS</b>	Data Description Specifications. A means of creating external file definitions on the AS/400. With DDS, you will define each field within the file you are defining. DDS are created using the AS/400 facility Source Entry Utility.
<b>DIF format</b>	*DIF. One of four micro file formats supported by ETU. DIF files are usually created with spreadsheet applications such as LOTUS 1-2-3.
<b>EBCDIC character set</b>	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.
<b>ETU</b>	Emulator Transfer Utility. AS/400 utility that transfers files to or from 5250 emulators. The best around.

<b>F&amp;I</b>	Format and Input specifications. A means of creating internal file definitions on the AS/400. ETU supports F&I specs.
<b>host system</b>	The AS/400 computer and its peripherals.
<b>IBMTRAN</b>	An interface program for PCs that is included with ETU. It allows non-ES32 emulation packages to be used with ETU.
<b>interactive processing</b>	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.
<b>job queue</b>	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.
<b>logical file</b>	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.
<b>OCI</b>	Object Constraint Language. A modeling language used to set up strings of valid OS400 commands.
<b>output queue</b>	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.
<b>physical file</b>	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.
<b>System/36 mode</b>	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.
<b>Tab format</b>	*TAB. One of four micro file formats supported by ETU. Tab files are usually created with data base applications. The fields in a TAB formatted file are separated by tab characters.
<b>Text format</b>	*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.

## Free Manuals Download Website

<http://myh66.com>

<http://usermanuals.us>

<http://www.somanuals.com>

<http://www.4manuals.cc>

<http://www.manual-lib.com>

<http://www.404manual.com>

<http://www.luxmanual.com>

<http://aubethermostatmanual.com>

Golf course search by state

<http://golfingnear.com>

Email search by domain

<http://emailbydomain.com>

Auto manuals search

<http://auto.somanuals.com>

TV manuals search

<http://tv.somanuals.com>