System Administrator’s Guide

cfSOFTWARE, Inc.
# Table of Contents

**Chapter 1 - Introduction**  
1

**Chapter 2 - Overview**  
3  
- Processing Summary.................................................................3  
- System Facilities........................................................................5  
  - File Transfer...........................................................................5  
  - User Exit...............................................................................5  
  - Data Extraction........................................................................6  
  - Data Formatting........................................................................7  
  - User Profiles...........................................................................7  
  - Central Control.........................................................................8  
  - Library Support.......................................................................8  
  - Performance and Integrity.......................................................9

**Chapter 3 - Getting Started**  
11  
- Introduction...............................................................................11  
- CICS - Function Walk-through................................................12  
  - pcMAINFRAME Menu............................................................12  
  - PC ID Directory....................................................................12  
  - PC ID Maintenance...............................................................13  
  - Profile Directory....................................................................15  
  - Profile Maintenance..............................................................16  
- TSO - Function Walk-through..................................................24  
  - System Form File....................................................................24  
  - Session Startup......................................................................24  
  - User ID Records.....................................................................25  
  - Profile Directory and Profile Maintenance............................27  
  - DDREC Records.....................................................................27

**Chapter 4 - Using the System**  
31  
- Online Maintenance Overview................................................31  
- Processing Conventions............................................................31  
- Functional Menus.....................................................................32  
- Online Maintenance..................................................................33  
  - Review Transmission Logs.....................................................33  
  - Profile Maintenance..............................................................35
### Chapter 5 - Managing pcMAINFRAME

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Administration</td>
<td>73</td>
</tr>
<tr>
<td>Security</td>
<td>73</td>
</tr>
<tr>
<td>Libraries</td>
<td>74</td>
</tr>
<tr>
<td>PC IDs</td>
<td>74</td>
</tr>
<tr>
<td>Backup and Recovery</td>
<td>74</td>
</tr>
<tr>
<td>Logging and Statistics</td>
<td>75</td>
</tr>
<tr>
<td>Controlling Profiles</td>
<td>75</td>
</tr>
<tr>
<td>Profile Definitions</td>
<td>75</td>
</tr>
<tr>
<td>Master and User Profiles</td>
<td>76</td>
</tr>
<tr>
<td>Profile Report - CFX750</td>
<td>76</td>
</tr>
<tr>
<td>Sample JCL</td>
<td>77</td>
</tr>
<tr>
<td>Options Configuration</td>
<td>80</td>
</tr>
</tbody>
</table>

### Chapter 6 - Host Initiation

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Initiation with CFXINIT</td>
<td>85</td>
</tr>
<tr>
<td>Overview</td>
<td>85</td>
</tr>
<tr>
<td>Transfer Startup</td>
<td>85</td>
</tr>
<tr>
<td>BATCH Data File Transfer</td>
<td>86</td>
</tr>
<tr>
<td>Record Length Considerations</td>
<td>86</td>
</tr>
<tr>
<td>KSDS VSAM Upload Considerations</td>
<td>86</td>
</tr>
<tr>
<td>Audit Files</td>
<td>87</td>
</tr>
<tr>
<td>Controlled Transfers</td>
<td>88</td>
</tr>
<tr>
<td>Simple controlled Transfers</td>
<td>88</td>
</tr>
<tr>
<td>Complex controlled Transfers</td>
<td>88</td>
</tr>
<tr>
<td>Return Codes</td>
<td>89</td>
</tr>
<tr>
<td>Startup/Communications Status</td>
<td>89</td>
</tr>
<tr>
<td>Controlled Transfer Completion Status</td>
<td>89</td>
</tr>
<tr>
<td>Batch Transfer Status</td>
<td>89</td>
</tr>
<tr>
<td>Return Code Testing - Batch transfers</td>
<td>90</td>
</tr>
<tr>
<td>Command File</td>
<td>91</td>
</tr>
<tr>
<td>Command Syntax</td>
<td>91</td>
</tr>
<tr>
<td>Multiple Requests</td>
<td>92</td>
</tr>
<tr>
<td>Command File Format</td>
<td>93</td>
</tr>
<tr>
<td>Parameter Descriptions</td>
<td>94</td>
</tr>
<tr>
<td>Program Processing</td>
<td>100</td>
</tr>
<tr>
<td>Process Reporting</td>
<td>100</td>
</tr>
<tr>
<td>Sample Reports</td>
<td>101</td>
</tr>
<tr>
<td>Program Execution</td>
<td>101</td>
</tr>
<tr>
<td>Job Control Language Requirements</td>
<td>102</td>
</tr>
<tr>
<td>Examples</td>
<td>103</td>
</tr>
</tbody>
</table>
# Chapter 7 - Library Processing

> **Overview**

> **Purpose**

> **Definition**

> **Security**

> **Control**

> **Batch Access**

> **Processing Capabilities**

- **Generic Transfers**
- **Library Time Stamp Control Processing Rules**
- **Library History Log Processing**

> **Public and Private Libraries**

> **Online Processing**

- **Profile Definition - LIB**
- **Assigning Libraries to PC Users**
- **Library Directory — CICS Only**
- **Online Maintenance Functions**

**Library Utility Program**

- **Command File**
- **VSE JCL Requirements**
- **MVS JCL Requirements**
- **VSE Examples**
- **MVS Examples**

**Appendix A - Btrieve® File Access**

- **Mainframe Requirements**
- **PC Requirements**
- **Download Processing**
- **Upload Processing**

**Appendix B - Valid Text Characters**

- **Standard Text Characters**
- **Language-specific Text Characters**
- **Customized Text Translation**
- **Spool Control Characters**
- **Performance Considerations**

**Appendix C - Coding User Exits**

- **Overview**
- **Control Block Definition**
- **Sample Exits - Data Base Access**

**Appendix D - VSE Power Access**

- **CFXPOWEX Exit Parms for Queue Transfers**
- **CFXPOWEX Exit Parms for Command Processing**
- **Generic Transfers**
- **Sample Profiles**
- **Upload/Download Specifications**
- **Useful VSE/POWER Commands**
Appendix E - JES Spool Access 173
  Downloading Reports ................................................................. 174
  Reducing Download Bottlenecks with TSLIMIT ...................... 174
  Uploading Reports .................................................................... 175
  Uploading to the Reader .......................................................... 175
  CFXJESEX ExitParms ................................................................. 175

Appendix F - CFX$FILE User Exit 179
  CFX$FILE .................................................................................. 179
  INDS$FILE ............................................................................... 179
  CFX$FILE Exit Parameters ....................................................... 180
  QNAME/HEADER Format .............................................................. 181
  CFX$FILE Sample Profile ............................................................ 182

Appendix G - Security Exits 185

Appendix H - BIM-Edit and VSE Library Access 191
  CFXBIMX ExitParms for Data Transfers ...................................... 191

Glossary of Terms 195

Index 203
pcMAINFRAME is a general purpose tool that links personal computers and mainframes flexibly to suit varying data transfer demands while insuring that data access always remains under central control. This manual is written for the person who will administer your pcMAINFRAME system at the mainframe. It assumes that the reader has an understanding of the data files to be used by pcMAINFRAME on the mainframe and some understanding of CICS or TSO, depending upon the version of pcMAINFRAME that will be installed. This manual describes the mainframe administration tasks in detail, and is organized as follows:

Chapter 1 - Introduction explains the organization of this manual and identifies other manuals that are available.

Chapter 2 - Overview is an introduction to pcMAINFRAME features.

Chapter 3 - Getting Started steps through several file maintenance features, such as PC IDs and profiles. It gives you an understanding of how the system works and prepares you for the detail covered by the rest of the manual.

Chapter 4 - Using the System can be used as a reference to pcMAINFRAME features, which are explained here in detail.

Chapter 5 - Managing pcMAINFRAME outlines the areas for which the administrator of pcMAINFRAME is directly responsible:

- security
- PC IDs
- Profile maintenance
- logging
- backup and recovery
- pcMAINFRAME configuration and libraries.

Chapter 6 - Host Initiation discusses the facility in pcMAINFRAME to initiate transfers from a batch job step on the host and optionally to perform data transfers directly from this batch step.

Chapter 7 - Library Processing defines the properties and features of the pcMAINFRAME library facility. It includes documentation on the use of the batch library utility program, CFXLIBU.
Appendix A - Btrieve File Access is for those users of pcMAINFRAME who wish to transfer data directly to and from their Btrieve Files.

Appendix B - Valid Text Characters lists the text file characters and optional translate tables for use during pcMAINFRAME transfers.

Appendix C - Coding User Exits is a guide to user exit coding and security.

Appendix D - VSE Power Access explains the use of the VSE POWER Spool access facility.

Appendix E - JES Spool Access explains the use of the JES Spool access facility.

Appendix F - CFX$FILE User Exit describes how to use the CICS Temporary Storage Queue access facility (IND$FILE) with pcMAINFRAME.

Appendix G - Security Exits explains how to provide customized security for all pcMAINFRAME maintenance and transfer functions.

Appendix H - BIM-Edit and VSE Library Access denotes how users, through the user exit CFXBIMX, can have access to the BIM-EDIT library or DOS/VSE sublibrary.

Additional information about pcMAINFRAME administration and use can be found in the following manuals:

pcMAINFRAME Installation Guide covers the details of installing pcMAINFRAME on your mainframe.

pcMAINFRAME Messages lists the messages that are issued by pcMAINFRAME and explains each of them. Suggested actions are also listed where appropriate.

pcMAINFRAME PC Administrator’s Guide describes the environment of the PC portion of pcMAINFRAME and the utilities used to configure and use pcMAINFRAME.

pcMAINFRAME PC User’s Guide describes the operation of pcMAINFRAME at the PC.

pcMAINFRAME Samples Section contains fully annotated examples of the setup files, profiles and dialog scripts needed for Host Initiation, Host batch transfer, Record selection, summarization, and reformatting, Advanced Compression and Multiple File Support.

cfSOFTWARE is committed to ensuring that pcMAINFRAME is a product that will continue to be of value. We invite your comments, suggestions, insights and questions as you work with pcMAINFRAME so that we can be responsive to your needs.

Our technical support number is (847) 824-7180.

Visit us at our web site: www.cfsoftware.com
Chapter 2 - Overview

Processing Summary

pcMAINFRAME is an intelligent file transfer system that selects, summarizes, reformats, and transfers data between PC and mainframe files over a wide variety of communications facilities. To accomplish this, pcMAINFRAME is installed on both the PCs and the mainframe. The software on each platform performs two distinct functions: one performs communications and data transfer, and the other defines and configures pcMAINFRAME facilities.

The communications programs on the mainframe and on the PC together create a cooperative processing environment. They both provide intelligence critical to the transfer task. pcMAINFRAME’s proprietary protocol processes requests, maintains security, compresses data, checks for errors, translates character codes and converts file and data formats between the mainframe and PC.

The mainframe programs are responsible for establishing and controlling security functions, defining data access, recording activity, selecting and converting data, and performing mainframe communications. Details of these operations are contained in this manual.

The PC programs are responsible for the user interface, options and communications configuration, data management on the PC, and communications from the PC. Detailed explanations of these functions are contained in the pcMAINFRAME PC Administrator’s Guide.

pcMAINFRAME users can select data from portions of mainframe files and records. Entire records or specific fields of selected records can be transferred. In addition, pcMAINFRAME can summarize detail records and then transfer the summary records.

As data is sent, it is automatically converted to the appropriate file format. Translation between EBCDIC and ASCII takes place, and field format (packed, binary, character) conversion is performed. The file structure itself is translated between mainframe format and one of several PC varieties: xBASE, worksheet, fixed, DIF, comma delimited, text and binary.
Throughout this manual, we will use several terms specific to pcMAINFRAME. Three basic terms which you will need to be familiar with are:

- **PC ID or User ID**: a unique identifier for each user of pcMAINFRAME. In CICS, it is any user-defined identifier. In TSO, it is always a 7-character TSO USERID.
- **Profile**: a description of data that is to be transferred by pcMAINFRAME. It describes the file, data type, record and field selection, and output format of transferred data. PC users access data by profile name.
- **Library**: a special mainframe file used by pcMAINFRAME to support storage of PC uploaded data. Libraries store and organize data on the mainframe by the original PC filename.

Other terms are defined in the Glossary at the end of this guide.

The mainframe portion of pcMAINFRAME consists of two primary functional systems: file maintenance processing (PCMM) and file transfer processing (PCMX). Each system is made up of several interrelated programs.

Figure 2-1 shows the relationship between the various programs on the PC and the mainframe. Following the diagram is a brief description of PCMM and PCMX. The use of these transactions is covered in subsequent sections of this manual. The rest of this chapter is devoted to describing pcMAINFRAME system facilities.

![Figure 2-1: pcMAINFRAME Control Flow](image-url)
**PCMX** is the communications transaction that performs data transfer and interfaces with the PC. The PCMX transaction performs all data access, selection, formatting, and communications on the mainframe. This transaction runs in CICS whenever pcMAINFRAME is communicating with a PC.

**PCMM** is a transaction that is used by the system administrator. It is used to create and maintain the profiles that control data transfers initiated by PC users. PCMM creates and maintains PC IDs, which control access to the system. It displays library directories and deletes library members. PCMM is also used to view the on-line mainframe activity log that pcMAINFRAME maintains.

---

**System Facilities**

This section provides brief descriptions of pcMAINFRAME conventions.

**File Transfer**

pcMAINFRAME’s primary purpose is to transfer data from a PC to the mainframe (an upload) or from the mainframe to a PC (a download). Data to be downloaded may be selected based upon key ranges of files and records meeting certain criteria. The user may then send the selected records or particular fields to the PC. pcMAINFRAME also has the capability of summarizing detail records and transferring these summaries.

pcMAINFRAME is capable of working with the following mainframe file types: VSAM KSDS, RRDS, ESDS, Library (part of the pcMAINFRAME system discussed in this chapter) and system spool files. For TSO users, access to PDS and QSAM data sets is also supported. The maximum fixed record length that can be transferred is 32,000 bytes.

PC file formats supported include: DIF, comma delimited, text, binary, worksheet, or xBASE. An optional PC file format available is Btrieve® file access.

As the data is sent to or from these files, it is automatically converted to the appropriate file format. EBCDIC to ASCII translation takes place and field format (packed, binary, character) conversion is performed. The file structure itself is translated between mainframe format and one of several PC varieties. See the Data Formatting section later in this chapter for additional conversion information.

This file transfer capability opens many new opportunities to PC users in a mainframe environment. Mainframe data that has been inaccessible or has required manual re-entry can now be transferred easily to the PC. The mainframe’s large storage capacity can be used by the PC for archival storage, backup, or a central distribution facility among many PCs. The PC can also become a data entry or data collection facility and subsequently upload data for processing by mainframe applications.

**User Exit**

An exit facility allows user application programs to interface with pcMAINFRAME. This exit is functional for data transfer in either direction and passes data at the record level. This permits users to extract, add, update and transfer information in data base files, multiple datasets or other file structures that are not supported by pcMAINFRAME. See Appendix C, *Coding User Exits* for additional detail.
A security exit is also available. This exit is invoked at startup and the beginning of each transfer to provide for the application of user specific security rules. See Appendix G, Security Exit for addition detail.

**Data Extraction**

One of pcMAINFRAME's most powerful features is its ability to extract, summarize, and reformat data before downloading it to the PC. The mainframe typically houses large volumes of detailed information that may not be useful to the PC user. pcMAINFRAME allows the user to reduce the amount and to extract specific bits of data for transfer through the use of profiles. Criteria for browsing and selection may be specified as either fixed or variable. Fixed criteria cannot be changed during file transfer, while variable criteria is specified when a transfer is started.

**BROWSE** - This function is applicable only to KSDS files. It allows you to restrict file access to selected ranges, limiting the number of mainframe records read to produce output to the PC. You can use it to select, for example, specific departments from a general ledger file or certain products from a sales file. The browse function searches by record key.

**RECORD SELECT** - Records within the mainframe file that match specified criteria can be included or excluded with this function. The criteria for records selected are defined at the field level, not by a key as is the case for the Browse function. Selection criteria are specified by giving the following:

- the field name
- the beginning location within the record
- the length of the field
- the data type: character, numeric, packed, or binary
- the number of decimal positions for numeric fields
- the function: include or exclude
- the comparison operator: =, >, <, >=, <=, <> , lo, and hi
- the value to be compared against.

**BREAK CONTROL** - This function is used to accumulate data from multiple mainframe records into a single record before download. Break control is used to specify conditions on which an output record will be created and sent to the PC. For example, if a file of accounts is sequenced by date within account and you wish to produce a summary of accounts by month, you would specify a break upon any change in month or account. This insures that the totals will correctly reflect the three possible situations:

- month changes but account number does not
- account number changes but month does not
- both change at the same time.

**OUTPUT RECORD FORMATTING** - This feature allows you to define which fields will be transferred and how they will appear in the output file. The order in which the fields are specified determines the order in which the fields will appear in the output record. Information specified for each field includes the following:

- the field name
- the beginning position of the record
- field lengths in bytes
Data Formatting

Data is translated between EBCDIC to ASCII automatically, unless transparent data format is specified. During translation, individual fields are checked for valid content, invalid characters are replaced with spaces or zeros, and a warning message is sent to the PC. When downloading, pcMAINFRAME formats data according to output file format. The ten types of file formats that pcMAINFRAME supports are: ASCII / Text, BASIC, Excel, DIF, Fixed, Include, Transparent (Binary), Spool, WKS, and xBASE. These data types cover the spectrum of PC applications data types.

User Profiles

User profiles are designed to allow pcMAINFRAME users to view and maintain their own profile records without compromising data security, and to provide an additional layer of selectivity for data extraction. While master profiles extract data from a physical file, a user profile extracts data from a "logical" file created by its master profile.

When accessing user profiles, PC ID and password are used to determine the operator class and limit access to profiles that have a matching class. A user profile specifies the name of its master profile and accesses only the records that meet the browse range, record select, and break control requirements of its master profile, and fields that are defined in the output format section of the master profile. User profiles provide the systems administrator with the ability to limit user access to a subset of a file, but also give the user full control over selection and format of those records and fields he is authorized to access.

Security

pcMAINFRAME provides several levels of security that insure controlled access to the system itself, as well as to data within the system. The first level of security is one inherent in the nature of pcMAINFRAME, its operating protocol. Should someone attempt to use the system and not follow this protocol, the mainframe program will immediately terminate the session. This makes it extremely difficult for a non-pcMAINFRAME user to break into the system.

The second level of security is on the individual PC user level. During the start up of a pcMAINFRAME file transfer, the individual user's ID is transmitted to the mainframe and is matched against a file of established IDs. The ID passed must be valid. Then, if the ID requires a password, the user is prompted for a password (up to three tries), and it too must match. If either the user’s ID or password is rejected, the session is terminated. This restricts pcMAINFRAME access to those users who have pcMAINFRAME software on their PC and are authorized users of pcMAINFRAME.

When the system is set up, maximum security and function may be insured by establishing a unique ID and password for each user, or access may be made common for groups of users by assigning them a single ID. At a given time only one user with a common PC ID may use the system.

The third level of security is provided by up to eight class codes that are defined for each ID. This level controls access to data more directly. Each profile can specify its
own password, and must specify an ID class authorized to access it. This ensures that only those
users with the appropriate class authorization, and optionally users who know the profile password
(not the same as their PC ID password), will have access to the data defined by a profile. Library data
may be similarly restricted at the time the library member is created or updated.

A final level of security provided is a "SECURITY EXIT" option. This option, when activated, will invoke a user-written security exit program whenever pcMAINFRAME functions (PCMM or PCMX) are invoked. The security exit has final authority over the user’s ability to perform a requested function.

Central Control

pcMAINFRAME places control of all functions in the hands of the system administrator. The following decisions are made and implemented centrally:

- which PCs have access to the system
- which profiles each user may access
- what data is to be accessed in each profile
- which direction each profile may transfer data.

PCs using the system cannot override or alter this control.

A file of valid PC IDs is built and maintained by the system administrator. Each PC ID is associated with a password and up to eight classes of profiles that the PC can access. These measures allow the administrator to restrict user access to certain records and certain fields.

Data is accessed by specifying a profile name. Profiles describe the file to be accessed, the mainframe file type, the data type that is to be created on the PC, its security class, an additional profile password when necessary, the transfer direction (upload or download), and any selection and formatting criteria. Because of class and password protection, the administrator can designate which users have access to certain profiles, while profiles themselves determine which ranges, record types, and fields the user can access.

Library Support

pcMAINFRAME provides a public and private library facility that allows users to store files from their PCs on a mainframe file. Any file may be uploaded to the library and is retained under the same name that it had on the PC. Libraries may be accessed and downloaded through a profile. However, members are accessed, not the entire dataset. Library members may be shared by users, turning this into a common facility for transferring programs and data between PCs.

Users may use this facility as:

- additional storage capacity,
- backup to PC storage,
- an archiving facility using mainframe backup capabilities,
- an input queue for data created on the PC being uploaded to a mainframe application,
- a means of distributing data or programs to many different PCs,
- a means of distributing common software,
- and a bulletin board service.
Performance and Integrity

pcMAINFRAME has several performance enhancing features. Data compression reduces the number of characters sent over the communications facility and reduces transmission time. Record selection, output formatting and summarization, and browsing reduce the quantity of data that is read and the number of records written to the PC. Restart facilities give the user the ability to resume a long transmission at the point of a communication failure rather than having to re-transmit an entire file. The system has the ability to recover from such "soft" errors as disk file overflow and temporary communication errors.

pcMAINFRAME operates in four communications environments: protocol converter, LU6.2, TCP/IP and 3270 emulation. In 3270, TCP/IP and LU6.2 modes the hardware performs many error detection and correction functions; but in protocol converter mode, neither the hardware nor the operating system provide these functions. pcMAINFRAME does. It uses several techniques to ensure that the data received is the same as the data transmitted. Each block of data sent between the mainframe and a PC is subject to parity checking, block sequence checking, and CRC checking. Any block that does not pass these tests is automatically retransmitted until it is correctly received or until the error threshold is reached and the transmission is terminated.
Chapter 3 - Getting Started

Introduction

This chapter is a walk-through of the file maintenance facilities of pcMAINFRAME. After pcMAINFRAME is installed, follow the steps outlined below from your terminal for a brief introduction to each of the file maintenance options in PCMM. After you have completed this exercise, you will know how to view and maintain the various records needed to administer pcMAINFRAME and understand the basic working of the system. A complete description of all fields and facilities is provided in Chapter 4 - Using the System.

Before you get started, you will need to understand the uses of the following keys:

- **Enter**: Pressing this key causes the application to edit all input on the current screen, produce any appropriate messages, and take appropriate action.
- **Clear**: Returns to the menu and returns the application to its initial status. Any entries keyed prior to entering **Clear** are ignored. If **Clear** is pressed while the menu is displayed, PCMM terminates and control passes back to CICS.
- **F1**: Returns to the initial screen of the sub-function in progress. Any updates or changes entered on the screen will be performed to the working copy.
- **F3/F4**: Moves the display to the next screen within a sub-function. Any updates or changes entered on the screen will be performed to the working copy.
- **F5/F6**: Moves the display to the previous screen within a sub-function. Any updates or changes entered on the screen will be performed to the working copy. If you enter an update function from a directory a **F5** will return you to the directory.

Although the TSO and CICS versions have similar System Administration functions, there are some differences. If you are running pcMAINFRAME for TSO, proceed to the section labeled ‘TSO - Function Walk-through’ later in this chapter.
CICS - Function Walk-through

pcMAINFRAME Menu

Sign on to CICS and enter the transaction ID, PCMM. This is the entry point into the mainframe functions of pcMAINFRAME. The first screen will be the pcMAINFRAME menu as shown in figure 3-1:

![Figure 3-1: pcMAINFRAME Menu](image)

The nine items on the menu are the functions used to manage pcMAINFRAME. They are used to list, view and maintain the two basic control records, PC ID and Profile records, or to view library directories and transmission logs. This walk-through will demonstrate each function in a logical progression so you can better understand the relationships between the various functions.

PC ID Directory

- Type 5, **PC ID Directory**, and press Enter

![Figure 3-2: PC ID Directory](image)
Initially, a blank PC ID Directory screen is displayed. The cursor is positioned in the ID field which accepts either full or partial PC IDs. Once you enter a PC ID and hit (Enter), existing PC IDs will display as shown in the figure above.

- Press [Enter].
  
The screen will display PC IDs starting from the beginning of the file.

- Type P and press [Enter].
  
The display will start with the first PC ID beginning with P. If you enter a full PC ID, the display will start with that ID.

Try various combinations to see how the display positions itself with different inputs.

- Use the [F3] or [Return] key to move the cursor down to one of the detail lines and then press [Enter].
  
The screen will now display a detailed description of the PC ID selected. At this point, if you wanted to view another PC ID, you would press [F3] to return to the **PC ID Directory** screen.

- Press the [Enter] key and the display will return to the Function Menu.

**PC ID Maintenance**

- Type 3, **PC ID Maintenance**, and press [Enter].

<table>
<thead>
<tr>
<th>(pcMAINFRAME)</th>
<th>PC ID Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC ID............DEMO</td>
<td>4 Characters. Identify a user to pcMAINFRAME</td>
</tr>
<tr>
<td>Description......THIS IS A DEMO</td>
<td>...Comments</td>
</tr>
<tr>
<td>Password............DEMO</td>
<td>1 to 8 character password for this ID.</td>
</tr>
<tr>
<td>Classes............A B C</td>
<td>1 to 8 classes A - Z. * = all profiles</td>
</tr>
<tr>
<td>Private library...CFXLIB</td>
<td>FCT name of this PC’s private library.</td>
</tr>
<tr>
<td>Public library...CFXLIB</td>
<td>FCT name of this PC’s public library.</td>
</tr>
<tr>
<td>Customized View...P</td>
<td>Customized User Profile VIEW authorization</td>
</tr>
<tr>
<td>Customized Save...*</td>
<td>Customized User Profile SAVE authorization</td>
</tr>
<tr>
<td>Authorized Prefix AB PC AC PA</td>
<td>Up to 4, for User Profile VIEW/SAVE</td>
</tr>
<tr>
<td>Max IDs in Pool..000</td>
<td>Maximum number of PCIDs in this Pool</td>
</tr>
<tr>
<td>Last maint date.. Date this PCID was last maintained.</td>
<td></td>
</tr>
<tr>
<td>Last used date... Date this PCID was last used.</td>
<td></td>
</tr>
</tbody>
</table>

Enter ID to be updated.

DEPRESS CLEAR TO RETURN

*Figure 3-3: PC ID Maintenance*

The **PC ID Maintenance** screen is displayed.

- Type DEMO and press [Enter].
  
  All information about this ID is shown on the screen.

- Return to the main menu by pressing [F3].
Add a New PC ID

- Type 5, PC ID Maintenance, and press Enter. The cursor should be in the PC ID field and the ID should be blank.
- You can enter a new ID of up to four characters. In this case, type TEST.
  
  If you typed four characters, the cursor will be in the DESCRIPTION field; if you typed fewer than four characters, press Tab.
- Type up to 35 characters for a description of this PC ID; type THIS IS A DEMO.
  
- Enter a password of up to eight characters to be associated with this ID; type MYPASS.
- 5 to the class fields and enter A B C , the classes of profiles that this PC ID is to have access to.
- 5 past the last five classes and the library fields since the defaults are correct.
- Press Enter
  
  If you have entered the information correctly the message "Add function in progress for this PC ID" will appear on screen. Press Enter. If you have entered information incorrectly, an error message will appear on screen. Fix the entry in error and then press Enter.
- Type A in the field at the bottom of the screen that says Enter Update Selection and then press Enter (the cursor will be in the PC ID field).
  
  The screen will return with the message "Addition of TEST was successful."
- Press Enter again
  
  The PC ID that you added will be displayed in full.

Update an Existing PC ID

To change the password, for example:

- 5 to the Password field and type Newpass.
- Hit Enter or tab to the Enter Update Selection field and type U. Press Enter
  
  The message "Update of TEST was successful" will appear.

You have now added and updated a PC ID. Try other updates and delete TEST when you are through by typing D on the Enter Update Selection line. When you are done, press the Clear key or Ret to return to the menu.
Profile Directory

- Type 4 for Profile Directory and press the Enter key.

A blank Profile Directory will be displayed. The cursor is in the Name field. This field will accept either a full or partial profile name to position the beginning of the directory list.

- Press Enter.

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>Class</th>
<th>Dataset</th>
<th>Type</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFXKSDS</td>
<td></td>
<td></td>
<td></td>
<td>Fixed</td>
<td>DOWNLOAD 123 DATA FOR TEMPLATE ABC</td>
</tr>
<tr>
<td>DEMO1</td>
<td>CFXKSDS</td>
<td>KSDS</td>
<td>Basic</td>
<td></td>
<td>DOWNLOAD DATA FOR Template ABC</td>
</tr>
<tr>
<td>LIB</td>
<td>CFXLIB</td>
<td>Lib</td>
<td>Trans</td>
<td></td>
<td>UPLOAD LIBRARY FILE</td>
</tr>
<tr>
<td>LIBTRAN</td>
<td>CFXLIB</td>
<td>Lib</td>
<td>Trans</td>
<td></td>
<td>TRANSPARENT MODE LIBRARY TRANSFER</td>
</tr>
<tr>
<td>LIBW</td>
<td>CFXLIB</td>
<td>Lib</td>
<td></td>
<td></td>
<td>UP OR DOWN LOAD LIB - WORD PROCESS</td>
</tr>
<tr>
<td>POWER01</td>
<td>CFXPOWEX</td>
<td>Exit</td>
<td>Spool</td>
<td></td>
<td>DOWNLOAD A REPORT FROM VSE/POWER</td>
</tr>
<tr>
<td>POWER02</td>
<td>CFXPOWEX</td>
<td>Exit</td>
<td>Spool</td>
<td></td>
<td>UPLOAD A REPORT TO VSE/POWER</td>
</tr>
<tr>
<td>POWER03</td>
<td>CFXPOWEX</td>
<td>Exit</td>
<td>Spool</td>
<td>Fixed</td>
<td>DOWNLOAD PUNCH FROM VSE/POWER</td>
</tr>
<tr>
<td>POWER04</td>
<td>CFXPOWEX</td>
<td>Exit</td>
<td>Spool</td>
<td>Fixed</td>
<td>UPLOAD PUNCH TO VSE/POWER</td>
</tr>
<tr>
<td>POWER05</td>
<td>CFXPOWEX</td>
<td>Exit</td>
<td>Spool</td>
<td>Fixed</td>
<td>UPLOAD JOBS TO VSE/POWER</td>
</tr>
<tr>
<td>POWER06</td>
<td>CFXPOWEX</td>
<td>Exit</td>
<td>Trans</td>
<td></td>
<td>UPLOAD BINARY FILE TO POWER PUN</td>
</tr>
<tr>
<td>POWER07</td>
<td>CFXPOWEX</td>
<td>Exit</td>
<td>Trans</td>
<td></td>
<td>DOWNLOAD</td>
</tr>
<tr>
<td>POWER08</td>
<td>CFXPOWEX</td>
<td>Exit</td>
<td>Fixed</td>
<td></td>
<td>DOWNLOAD A QUEUELIST FROM VSE/POWER</td>
</tr>
<tr>
<td>POWER09</td>
<td>CFXPOWEX</td>
<td>Exit</td>
<td>Fixed</td>
<td></td>
<td>ISSUE DISPLAY CMD/DOWNLOAD RESULTS</td>
</tr>
<tr>
<td>SALES1</td>
<td>CFXKSDS</td>
<td>KSDS</td>
<td>DIF</td>
<td></td>
<td>DEMO - SALES BY BRANCH <em>TEST</em></td>
</tr>
</tbody>
</table>

Figure 3-4: Profile Directory

The screen will display profiles beginning with the first entry on the file. Figure 3-4 is an example of a display similar to the one you will see. You may use the Tab or Return keys to position the cursor to a line containing a profile and press Enter to go into update mode for that profile.

- Press Enter to return to the menu.
Profile Maintenance

Profile records define the content of the files in a given transfer.

In the following example, you will create a representative profile from start to finish. When you finish, the profile you have defined may be used as a guide for future profiles.

The profiles you create will access data in the file CFXKSDS, shipped with your pcMAINFRAME software. The table below gives the record layout of the file CFXKSDS that we will be using.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Begin Position</th>
<th>Data Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region</td>
<td>ZD</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>District</td>
<td>ZD</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Branch Office</td>
<td>ZD</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Product Number</td>
<td>CH</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Sequence</td>
<td>Bin</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>Month</td>
<td>ZD</td>
<td>17</td>
<td>2</td>
</tr>
<tr>
<td>Day</td>
<td>ZD</td>
<td>19</td>
<td>2</td>
</tr>
<tr>
<td>Year</td>
<td>ZD</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td>Dollar Sales</td>
<td>PD</td>
<td>23</td>
<td>6</td>
</tr>
<tr>
<td>Unit Sales</td>
<td>PD</td>
<td>29</td>
<td>3</td>
</tr>
<tr>
<td>Description</td>
<td>CH</td>
<td>32</td>
<td>30</td>
</tr>
<tr>
<td>Billing Code</td>
<td>CH</td>
<td>62</td>
<td>1</td>
</tr>
</tbody>
</table>

The first 16 bytes, fields Region through Sequence, are the key to this file. The profile that will be created in the following examples use this layout and produce a download with extraction and summarization. Figure 3-5 shows what the screen will look like when you are finished.
**Profile Definition**

Begin from the pcMAINFRAME Menu.

- Type 2, Profile Maintenance, and press the \[Enter\] key to get to the *Profile Definition* screen.

- Type the following inputs in their corresponding fields, tabbing past any fields in which the default value is the desired value:

  - **Profile name:**  SAMPLE
  - **Profile description:** *This is a sample download*
  - **Password:**  DEMOPASS
  - **Profile class:**  A
  - **File or Exit Name:**  CFXKSDS
  - **File Type:**  KSDS
  - **Output data format:**  F
  - **Transfer direction:**  D
  - **Generate column headers:**  N
  - **Translation modes:**  W
  - **Read limit:**  800
  - **Write Ctrl_Z at EOF:**  Y
  - **Write limit:**  0
  - **Make blank fields Null:**  N
  - **Skip type:**  B
  - **Count:**  0
  - **Multiple record Header:**  N
  - **Fixed rec length:**  leave blank

- Press the \[Enter\] key. pcMAINFRAME will edit the input and display any errors.

- Press the \[F3\] key to advance to the *Access Ranges and Limits* screen.

```
Profile: SAMPLE
Profile Definition
Access Ranges and Limits

<table>
<thead>
<tr>
<th>Data/Field Name</th>
<th>Browse Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGION1</td>
<td>01</td>
</tr>
</tbody>
</table>

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete.
PF1 - 1st page, PF2 - Next Page, PF3 - Prior Page, CLEAR - Menu
```

Figure 3-6: Access Ranges and Limits

For a KSDS file, up to 10 defined browse ranges can be read by pcMAINFRAME, thus limiting the portions of the file read and improving response time. In this screen, we will restrict this profile to accessing only the first region.

- Type **Region 1** into the **Data/Field Name** field and tab to the **From** field. Type **01**. Tab to the **To** field and type **01**. Press \[Enter\] and look for the message "All data entered is valid."
Profile Definition - Multiple File Definition

- Press \[ \text{PR} \] to advance the display to the Multiple File Definition screen. This screen can be used to define additional VSAM files to be accessed during a download. This example does not define multiple download files.

Profile Definition - Record Selection

- Press \[ \text{PR} \] to advance to the Record Selection screen.

### Profile Definition - Record Selection

#### Table 3-7: Record Selection

<table>
<thead>
<tr>
<th>Data/Field Name</th>
<th>Begin Pos</th>
<th>Bytes</th>
<th>Data Type</th>
<th>Dec Plc</th>
<th>Action</th>
<th>If</th>
<th>Comparison Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMOUNT</td>
<td>23</td>
<td>6</td>
<td>P</td>
<td>2</td>
<td>E</td>
<td>&lt;</td>
<td>500.00</td>
</tr>
<tr>
<td>DATE</td>
<td>17</td>
<td>6</td>
<td>D</td>
<td>I</td>
<td>LO &amp; FROM DATE (MMDDYY)</td>
<td>HI &amp; TO DATE (MMDDYY)</td>
<td></td>
</tr>
</tbody>
</table>

Define Include/Exclude rules . . . from none to 12 are allowed.

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete.

PF1 - 1st page, PF2 - Next Page, PF3 - Prior Page, CLEAR - Menu

Figure 3-7: Record Selection

This screen is used to allow pMAINFRAME to select records based on values contained in the data records. We will make entries that allow the PC user to select the date range for the download and to limit the download to records with more than $500 in sales. Figure 3-7 shows the completed screen.

Position the cursor on the input line in the column labeled Data/Field Name.

- Type the following inputs into the Record Selection screen using the \[ \text{GOB/G33/G29/G15/G0C} \] key to move from field to field:
  - Data/Field Name: Amount
  - Begin Pos: 23
  - Bytes: 6
  - Data Type: P
  - Dec Plc: 2
  - Action: E
  - If: <
  - Comparison Value: 500.00

  (to excludes all records with values of less than $500 in dollar sales).
Data/Field Name: Date
Begin Pos: 17
Bytes: 6
Data Type: D
Dec Plc: past this field
Action: I
If: LO
Comparison Value: &FROM DATE

Data/Field Name: past this field
Begin Pos: past this field
Bytes: past this field
Data Type: D
Dec Plc: past this field
Action: past this field
If: HI
Comparison Value: &TO DATE
(to select records within a date range entered by the PC user at the time the download occurs)

• Press and check for any error messages

**Profile Definition - Break Control**

• Press to proceed to the Break Control screen shown in Figure 3-8:

### Profile Definition - Break Control

<table>
<thead>
<tr>
<th>Data/Field Name</th>
<th>Begin Pos</th>
<th>Bytes</th>
<th>Data Type</th>
<th>Dec Plc</th>
<th>Action</th>
<th>If</th>
<th>Comparison Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGION</td>
<td>1</td>
<td>2</td>
<td>N</td>
<td>ANY</td>
<td></td>
<td></td>
<td>(Any Break In Field)</td>
</tr>
<tr>
<td>DISTRICT</td>
<td>3</td>
<td>2</td>
<td>N</td>
<td>ANY</td>
<td></td>
<td></td>
<td>(Any Break In Field)</td>
</tr>
<tr>
<td>BRANCH</td>
<td>5</td>
<td>4</td>
<td>N</td>
<td>ANY</td>
<td></td>
<td></td>
<td>(Any Break In Field)</td>
</tr>
<tr>
<td>PRODUCT</td>
<td>9</td>
<td>6</td>
<td>C</td>
<td>ANY</td>
<td></td>
<td></td>
<td>(Any Break In Field)</td>
</tr>
</tbody>
</table>

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete.
P1 - 1st page, PF2 - Next Page, PF3 - Prior Page, CLEAR - Menu

*Figure 3-8: Break Control*

This screen controls the time at which summary records are written. Up to six control fields may be specified. When the specified break condition is met, a summary record is written to the PC. Place the following entries into the Break Control screen:

Data/Field Name: Region
Begin Pos: 1
Bytes: 2
Data Type: N
Break If: Any

Data/Field Name: District
Begin Pos: 3
Bytes: 2

---

**System Administrator's Guide** Chapter 3 - Getting Started • 19
Chapter 3 - Getting Started System Administrator’s Guide

Data Type: N
Break If: Any

Data/Field Name: Branch
Begin Pos: 5
Bytes: 4
Data Type: N
Break If: Any

Data/Field Name: Product
Begin Pos: 9
Bytes: 6
Data Type: C
Break If: Any

Note: in Figure 3-7, the Comparison Value fields were entered manually by the user whereas in Figure 3-8, "(Any break in field)" was filled in automatically by pcMAINFRAME.

- Press [G0B/G28/G51/G57/G48/G55/G0C] and check for errors. The entries we just made will produce a summary by product, within branch, district, and region.

- Press [PF2] to advance to the next screen.

<table>
<thead>
<tr>
<th>Data/Field Name</th>
<th>Begin Pos</th>
<th>Bytes</th>
<th>Data Type</th>
<th>Break If</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch</td>
<td>5</td>
<td>4</td>
<td>N</td>
<td>Any</td>
</tr>
<tr>
<td>Product</td>
<td>9</td>
<td>6</td>
<td>C</td>
<td>Any</td>
</tr>
</tbody>
</table>

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete. PF1 - 1st page, PF2 - Next Page, PF3 - Prior Page, Clear - Menu.

Figure 3-9: Output Format

The Output Format screen specifies the format of the record sent to the PC and the order in which the fields will appear on the PC. For example, the Product Field will be first in our PC file even though it is the third in the mainframe file. Figure 3-9 shows what the screen should look like when completed.

Let’s define the records that will be sent to the PC. We are free to format this in any way we want. Below, the information that this profile will define and send to the PC is listed in order:

- Product Number
- Total Sales
- Region
- District
- Branch Office
- Average Sales
- Largest Sale

Output Format

Data/Field   Begin  ___MF____ ___PC____ Dec
Name       Pos   Size Type Size Type Plc Operation Picture / Inserted Data
PRODUCT          9     6 C       6          LAST      Alpha/Num X(6)
SALES           23     6 P       6        2 SUM       Packed S9(9)V9(2)
REGION          1     2 N       2          LAST      Numeric S9(2)
DISTRICT         3     2 N       2          LAST      Numeric S9(2)
BRANCH          5     2 N       2          LAST      Numeric S9(4)
AVG SALE        23     6 P       6        2 AVERAGE   Packed S9(9)\#9(2)
LARGEST         23     6 P       6        2 MAXIMUM   Packed S9(9)\#9(2)
Because break control has been defined, one record will be generated whenever a change in Region, District, Branch or Product occurs. You will notice that the data being sent to the PC is not in the same order as the fields in the input record are, and that some of the fields, such as Average Sales and Largest Sale, are derived by pcMAINFRAME.

Enter the data specified below into the *Output Format* screen. Each line of data corresponds to a line on the input screen.

<table>
<thead>
<tr>
<th>Data/Field Name</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>Product</td>
<td>Product</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Begin Pos</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mainframe Size</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mainframe Type</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decimal Places</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>Last</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picture/Inserted Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data/Field Name</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td>Sales</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Begin Pos</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mainframe Size</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mainframe Type</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decimal Places</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>Sum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picture/Inserted Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data/Field Name</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Region</td>
<td>Region</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Begin Pos</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mainframe Size</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mainframe Type</td>
<td>n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decimal Places</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>Last</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picture/Inserted Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data/Field Name</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch</td>
<td>Branch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Begin Pos</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mainframe Size</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mainframe Type</td>
<td>n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decimal Places</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>Last</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picture/Inserted Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Data/Field Name: Avg Sale  
Begin Pos: 23  
Mainframe Size: 6  
Mainframe Type: P  
PC Size: past this field  
PC Type: past this field  
Decimal Places: past this field  
Operation Average  
Picture/Inserted Data:  

Data/Field Name: Largest  
Begin Pos: 23  
Mainframe Size: 6  
Mainframe Type: P  
PC Size: past this field  
PC Type: past this field  
Decimal Places: past this field  
Operation Maximum  
Picture/Inserted Data:  

- Press Enter and check for errors. The PC Size is filled in to match the size of the field on the mainframe. Notice that we overlooked the field District.

- Using the Insert key, position the cursor to the R in Region and type over it with >, then press Enter.

  A blank line is inserted following the line containing Region. This insertion facility is available on all profile screens, pages two through five.

- Fill in the parameters for the new line:

  Data/Field Name: District  
  Begin Pos: 3  
  Mainframe Size: 2  
  Mainframe Type: N  
  PC Size: past this field  
  PC Type: past this field  
  Decimal Places: past this field  
  Operation Last  
  Picture/Inserted Data:  

- Press Enter and check for errors.

- Press PF1 to return to page 1 of the Profile Definition.

- Type A into the update control field and press Enter.

  The return message "Addition was successful for profile SAMPLE" indicates that you have just created your first profile.
Adding a Profile from a Model

- Press G0B/G28/G51/G57/G48/G55/G0C once again and the newly created profile will be displayed, ready for update.
- Tab to the field Profile Name and type over SAMPLE with SAMPLE2.
- Press G0B/G28/G51/G57/G48/G55/G0C

Note: the first message reads "Profile not found--Add Assumed". This indicates that we are now creating SAMPLE2 but we have retained all the attributes of SAMPLE. This allows modeling a new profile from an existing one.

- Press twice.

The Record Selection screen will be displayed and the criteria that you entered for SAMPLE are still present.

- Use the G0B/G37/G44/G45/G0C or arrow keys to place the cursor on the first unused line (after HI) and type:

  Data/Field Name: Units
  Begin Pos: 29
  Bytes: 3
  Data Type: P
  Dec Place: past this field
  Action: E
  If: <=
  Comparison Value: 5

- Press G0B/G28/G51/G57/G48/G55/G0C and then G0B/G33/G29/G14/G0C to return to page one.
- Type A and press G0B/G28/G51/G57/G48/G55/G0C.

You have added SAMPLE2 as a separate profile.

Sample Profiles

CFXFORM, the file that contains the profiles, was shipped to you containing numerous sample profiles. Please refer to them for examples of the various functions that may be performed with profiles.
TSO - Function Walk-through

System Form File

pcMAINFRAME uses a master file, known as the system form file, to carry specific information needed to control processing and run file transfers. The file is a VSAM KSDS file and consists of the following types of records:

- **OPTIONS** - contains configurable processing options and specific authorization/control data.
- **USER-ID** - contains information about a specific pcMAINFRAME user.
- **PROFILE** - contains the extract and format instructions for a specific transfer.
- **DDREC** - contains data set allocation information for all transfers.

This file is created and loaded when pcMAINFRAME is initially installed.

In addition to the systems form file, one or more individual form files can be defined to carry file transfer definition (PROFILE and DDREC) records. Individual form files can be defined at any time by an authorized pcMAINFRAME user. OPTIONS and USER-ID records are not carried or maintained on an individual form file.

The PCMM function is used to maintain both systems and individual form files. A CLIST named **PCMM** is used to run the file maintenance program CFX000T. Normally, the system form file is dynamically allocated and accessed during file maintenance. However, if a file with DDNAME=CFXIFRM is pre-allocated to the TSO session (either in the CLIST or anytime prior to running CFX000T), then the allocated file will be used during the maintenance session.

The following defines the records that can be viewed or maintained during file maintenance:

<table>
<thead>
<tr>
<th>Systems Form file:</th>
<th>UPDATE authorization</th>
<th>All functions are available</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>READ authorization</td>
<td>All functions, view mode only</td>
</tr>
<tr>
<td>Individual Form file:</td>
<td>UPDATE authorization</td>
<td>Profile and DDREC maintenance only</td>
</tr>
<tr>
<td></td>
<td>READ authorization</td>
<td>No functions</td>
</tr>
</tbody>
</table>

Session Startup

To initiate a maintenance session, sign on to TSO and run the CLIST, **PCMM**. The loadlib containing program CFX000T must be available to the TSO session. This will display the pcMAINFRAME maintenance Function Menu, as shown in figure 3-10:
The six items on the menu are the functions used to manage pcMAINFRAME. They are used to list, view and maintain the User ID, Profile and DDREC records. This walkthrough will demonstrate each function in a logical progression so you can better understand the relationships between the various functions.

**User ID Records**

The User ID record must be defined for each TSO ID that will use pcMAINFRAME file transfer. This record carries individual authorization information and is accessed before any file transfer is initiated. This record is not carried on an individual form file.

**User ID Directory**

The User ID Directory provides a summary list of all current ID’s defined to the system.

- Type **Enter**, *User ID Directory*, and press **Enter**

Initially, a blank *User ID Directory* screen is displayed. Key in a full or partial User ID and press enter. A summary listing of authorized ID’s, as shown in figure 3-11 will be displayed.

- Press **Enter** with spaces in the ID field.
  The screen will display IDs starting from the beginning of the file.
- Type **P** and press **Enter**.
The display will start with the first ID beginning with P. If you enter a full ID, the display will start with that ID.

- Use the or key to move the cursor down to one of the detail lines and then press .

The User ID maintenance screen appears. At this point you could make changes to the selected ID or press to return to the User-ID directory screen.

- Press the key and the display will return to the Function Menu.

**User ID Maintenance**

User ID Maintenance is used to add, change and delete User ID’s as needed.

- Type User ID Maintenance, and press .

  The User ID Maintenance screen is now displayed. The full key of the ID to be added or changed must be entered in the User ID field.

- Type TEST001 and press .

  All information about this ID is shown on the screen. Remember, you can also reach this point by selecting an ID from the User ID directory list.

**Adding a new User ID**

<table>
<thead>
<tr>
<th>pcMAINFRAME</th>
<th>User ID Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID........ TEST001</td>
<td>7 Characters. Identify a user to pcMAINFRAME</td>
</tr>
<tr>
<td>Description...... THIS IS A TEST ID</td>
<td>...Comments</td>
</tr>
<tr>
<td>IndForm File OK.. Y</td>
<td>Yes=Use for Individual form file only</td>
</tr>
<tr>
<td>Classes............. A B C</td>
<td>1 to 8 classes A - Z. *= all profiles.</td>
</tr>
<tr>
<td>Private library... CFXLIB</td>
<td>DNAME name of this PC’s private library.</td>
</tr>
<tr>
<td>Public library... CFXLIB</td>
<td>DNAME name of this PC’s public library.</td>
</tr>
<tr>
<td>Last maint date.. 12/15/99</td>
<td>Date this ID was last maintained.</td>
</tr>
<tr>
<td>Last used date... 00/00/00</td>
<td>Date this ID was last used.</td>
</tr>
</tbody>
</table>

UPDATE function in progress for this ID.

Enter "D" to delete or "U" to update ID.

Figure 3-12: User ID Maintenance

- Position the cursor in the User ID field
- Type in a valid TSO User-ID. For this demonstration, you may use TEST001. Press 

  The message "Add function in progress..." will appear in the message area. The cursor will be positioned in the Description field.

- Type a short description (up to 35 characters) of the User ID. Enter "This is a Test ID"

- Tab to the IndForm OK field and enter Y.

- Enter A B C in the class field to show the classes of profiles that this User ID is to have access to.

- Since the defaults for the library fields will be used, all data entry is complete. Press to start the edit process.
• Type A in the field at the bottom of the screen following the message "All data is valid, Type A to add ID" and press (Enter). Note: if any data is invalid, an error message will be displayed and must be fixed before adding the record.

• Press (Enter) again. The User ID that you added will be displayed in full. The screen will return with the message "Addition of TEST001 was successful"

Updating a User ID

• Tab to the Description field and change the description text. When all changes are complete, press (Enter) to start the edit process

• If all data is valid, the cursor will be positioned in the update Action field. Type U and press (Enter) to update the record. The message "Update of TEST001 was successful" will appear.

Deleting a User ID

• Type the full key of the User ID to be deleted TEST001 and press (Enter).

• Tab to the update action field at the bottom of the screen and type D. Press (Enter). The message "Deletion of TEST001 was successful" will appear.

You have now completed the User ID maintenance walk-through. Press the (Blank) key or (PF3) to return to the menu.

Profile records are the core of the pcMAINFRAME transfer system. They are used to define the mainframe record layout, any select/extract criteria used during download, and the type and format of the PC file that is to be transferred. The profile directory provides a list of all profiles defined on the FORM file being maintained.

Profile Directory and Profile Maintenance

The profile directory and maintenance screens used by pcMAINFRAME for TSO function just like the corresponding screens in pcMAINFRAME for CICS. Please see the Profile Directory and Profile Maintenance sections of the CICS Function Walk-through earlier in this chapter.

DDREC Records

DDREC records carry complete file allocation information for datasets that are accessed or created during file transfer. For existing files, only a DDNAME and full DSNAME are needed. For uploads to new sequential files, space and DCB parameters are also needed. This information is identical to the data on a JCL DDREC.

The key to this record is a seven character identifier that is assigned by the user. This identifier is used in profile records when specifying a file for data transfer.

DDREC Directory

The DDREC directory is used to list all defined DDNAMES and their associated DSNAMES that can be used during file transfer.

• Type (Blank), DDREC Directory, and press (Enter).

Initially, a blank DDREC Directory screen is displayed. Key in a full or partial DDREC-ID and press enter. A summary listing of assigned DDNAMES and DSNAMES, as shown in figure 3-13 will be displayed.
Figure 3-13: DDREC Directory

• Press \ with spaces in the ID filed.
  The screen will display from the beginning of the file.

• Type P and press Enter.
  The display will start with the first DDNAME beginning with P.

• Use the \ or \ key to move the cursor down to one of the detail lines and then press Enter.
  The DDREC Maintenance screen for the selected ID is displayed for possible update. To return to the directory list, press \.

• Press the \ key to return to the Function Menu.

**DDREC Maintenance**

DDREC Maintenance is used to add, change and delete the DDRECs that are used by pcMAINFRAME.

• Type 0, DDREC Maintenance, and press Enter.
  The DDREC Maintenance screen is displayed. The full key of the record to be added or changed must be entered to start processing.

• Type QSAM01 and press Enter.
  All of the information to be used for file allocation is displayed.

**Adding a new DDREC Record**

To assist in speeding maintenance, PCMM makes it easy to add a new DDREC by cloning an existing DDREC. With the cursor in the current position in the DDREC name field,

• Type in a new DDREC identifier. For this demonstration, use QSAM02. Press Enter.

• The message "Add function in progress..." will appear in the message area. The cursor will be positioned at the beginning of the DDREC field. Were you creating a new DDREC without cloning, the cursor would be positioned in the DSNAME field.
• Tab to the description field. Press \[PG] to clear the field. Type ‘This is a test new DDREC’. Now tab to the DSNAME field, and change the end of the dataset name to QSAM02 from QSAM01. Press Enter.

The remainder of the fields on the screen are used during upload processing only. They define the action to be taken if a file already exists or does not exist and the space, device, and DCB information needed to create the file.

Since this is an existing file that will be used for downloading, these fields need not be valued.

• If all data is valid, the cursor will be positioned in the Add Action field. Type A and press Enter to add the record. The message "Addition of QSAM02 was successful" will appear.

**Updating a DDREC**

• The cursor should be in the DDREC field, with QSAM02 still in the field. Press Enter. The data for QSAM02 that you just added will appear.

• Tab to the Description field and change the description text. When all changes are complete, press Enter.

• If all data is valid, the cursor will be positioned in the update Action field. Type U and press Enter to update the record. The message "Update of QSAM02 was successful" will appear.

**Deleting a DDREC**

• Type the full key of the DDREC to be deleted QSAM02 and press Enter.

• Tab to the action field at the bottom of the screen and type D. Press Enter. The message "Deletion of QSAM02 was successful" will appear.

You have now completed the DDREC maintenance walk-through. Press the [Backspace] key or [Esc] to return to the menu.
Chapter 4 - Using the System

Online Maintenance Overview

Processing Conventions

pcMAINFRAME administration is menu driven and operates under the control of CICS or TSO. Although transfers are initiated from the PC, mainframe administration governs all pcMAINFRAME transfers.

This chapter describes how to control end users and limit access to data by using administration facilities. Each of these functions is defined and presented from an administrator’s point of view.

The administration facility is entered by invoking a CICS transaction or a TSO CLIST. The default transaction ID in CICS and CLIST name in TSO for this function is PCMM. Note: this ID may have been altered by your systems programmer at installation time. If so, make note of the new ID and use it in place of PCMM whenever PCMM is referenced.

All of the screens in PCMM share common control conventions. Until one of the entries listed below is made, changes made to a file on screen will not be made to the master file. Returning to the Function Menu will remove all changes unless you enter:

- to add
- to delete, or
- to update.

There is also a common convention for editing data after fields are entered or changed and the \( \text{Enter} \) key is pressed. The pcMAINFRAME system will analyze the input and issue any warning or error messages as appropriate. The cursor is placed in the field or on the line requiring action so that the operator’s time is minimized in correcting the error.
Common control key conventions are described below:

- **/G0B/G28/G51/G57/G48/G55/G0C**
  - Pressing this key causes the application to edit all input on the current screen, produce any appropriate messages, and take appropriate action.

- **/G0B/G26/G4F/G48/G44/G55/G0C**
  - Returns to the menu and returns the application to its initial status. Any entries keyed prior to entering **/G0B/G26/G4F/G48/G44/G55/G0C** are ignored. If **/G0B/G26/G4F/G48/G44/G55/G0C** is pressed while the menu is displayed on the screen PCMM terminates and control passes back to CICS.

- **/G0B/G33/G29/G14/G0C**
  - Returns to the initial screen of the sub-function in progress. Any updates or changes entered on the screen will be performed to the working copy.

- **/G0B/G33/G29/G15/G0C** / **/G0B/G33/G29/G1B/G0C**
  - Moves the display to the next screen within a sub-function. Any updates or changes entered on the screen will be performed to the working copy.

- **/G0B/G33/G29/G16/G0C** / **/G0B/G33/G29/G1A/G0C**
  - Moves the display to the previous screen within a sub-function. Any updates or changes entered on the screen will be performed to the working copy. If you enter an update function from a directory a **/G0B/G33/G29/G16/G0C** will return you to the directory.

- **/G0B/G33/G24/G14/G0C** - **/G0B/G33/G24/G16/G0C**
  - Same as **/G0B/G26/G4F/G48/G44/G55/G0C**.

**Functional Menus**

The initial screen produced by PCMM is displayed in Figure 4-1 (CICS) or 4-2 (TSO). This is the menu which allows a choice of the desired sub-function.

![Figure 4-1: CICS pcMAINFRAME Menu](image-url)
Online Maintenance

Review Transmission Logs

**TSO Only**

The transmission logs for TSO are stored in a PDS. The name of the PDS is defined in the pcMAINFRAME Options record. The member name given to each log is the user’s TSO ID.

**CICS Only**

This facility allows you to view statistics and messages for the most recent file transfer session of a user. This log is reset each time an ID initiates a transfer session; a PC that is in the process of transmitting will have only a partial log available. A separate hardcopy log contains all messages that are displayed on the on-line log in chronological sequence. Each line is identified with the ID of the user receiving or sending the message.
Figure 4-3: Transmission Log Display

The PC ID field in the upper left hand corner will be blank. Type the ID of the PC that you wish to display and press \[ /G0B/G28/G51/G57/G48/G55/G0C \]. Any existing log messages for that PC ID will be displayed on the screen. The control keys (see the beginning of this chapter) will allow you to move through the log messages. Placing the cursor on a line in the transmission log and pressing the page forward key \[ /G0B/G33/G29/G15/G0C \] or \[ /G0B/G33/G29/G1B/G0C \] will position the selected line to the first line on the display. Placing the cursor on a line in the log and pressing the page backward key \[ /G0B/G33/G29/G16/G0C \] or \[ /G0B/G33/G29/G1A/G0C \] will position the selected line to the last line on the display. To display another PC ID enter the new ID in the PC ID field in the upper left hand corner and press \[ /G0B/G28/G51/G57/G48/G55/G0C \].

If History Logging has been activated, then a 'Y' in the History field will display historical log records. Note, log records are kept for a limited amount of time as specified in the Options record.

The following explains each line of the Transmission Log:

- **Mainframe/PC Versions:** the pcMAINFRAME mainframe software version followed by the PC software version.
- **Connect Date/Time:** the date on which the transmission was started including the day of the week and time of day.
- **Terminal ID/Task Nbr:** the CICS Terminal ID of the terminal that performed the transmission, followed by the CICS Task Number of the transmission.
- **PC ID:** the ID of the PC making the request.
- **Req #:** the Request Number(s) assigned to each transmission. One request number will be assigned sequentially for each upload or download. Other requests submitted by a PC will also be logged. The data associated with each is:
  - **Profile:** the Profile used for this request.
  - **PC file:** the name of the file to upload from or download to.
Uploaded/Downloaded Recs: the number of records transmitted during this request.

Req Duration (Request Duration): the elapsed time of the request in hours, minutes and seconds.

Bytes: the number of bytes transmitted in the request.

Char/Min (Characters/Minute): the transfer rate for this transmission.

Disconnect Date/Time: the date on which the transmission terminated including the day of the week/the time of day that the transmission terminated.

Connect Duration: the elapsed time of the transmission in hours, minutes and seconds.

Records Uploaded: the total number of records uploaded in this transmission.

Bytes Uploaded: the total number of bytes uploaded in this transmission.

Records Downloaded: the total number of records downloaded in this transmission.

Bytes Downloaded: the total number of bytes downloaded in this transmission.

Requests Processed: this is a count of the number of separate upload and download requests submitted during this transmission.

Profile Maintenance

Profiles are of key importance in pcMAINFRAME operation because the attributes defined by them determine how data is to be transferred. Profiles can be created, modified, deleted, displayed or modeled with the set of screens documented in this section.

Profile Definition Screen

The screen shown in Figure 4-4 is the first Profile Definition screen. When it is initially invoked, the first field, Profile Name, is blank. If you wish to update, model from, or display an existing profile, the profile name must be entered and the key pressed. This will cause a read of the file containing profiles and, if it is present, the requested profile will be displayed.
Figure 4-4: Profile Definition Initial Screen

Information for the following fields is entered on page 1 of the Profile Definition screen:

**Profile Name:** a one to eight character name identifies the profile. This field is **required** and must be unique. If a name is entered that already exists, pcMAINFRAME assumes that an update is being performed on the existing profile.

**Profile Description:** this **optional** field associates a one to thirty character description with the profile. This description appears on profile displays performed by PC users, and on the Profile Directory.

**Password for this Profile:** this field is **required** and may contain a one to eight character password that the PC user must enter when accessing this profile, or the word "NONE" indicating that no password is required from the user.

**Profile Class:** this field is **required** and may contain a single alpha character (A-Z) or an asterisk (*). This indicates a class to which the profile belongs and restricts the use of this profile to the PC IDs with authorization for this class. The asterisk (*) indicates that the profile may be used by any PC ID regardless of class.

**File or Exit Name:** This field is **required**.

**CICS:** This one to seven character name is the **FCT entry name** of the file that this profile will use or the **PPT name** of the user exit program. **Note:** the SERVREQ parameter of the FCT entry for this file must include the "BROWSE" as a valid function. If the file type is ESDS and uploading is to be performed, the FCT should define the RECFORM as VARIABLE BLOCKED to avoid record length errors.

**TSO:** This is the ddname of the dataset that will be used for transfer or the program name of the user exit program. If the ddname is preceded by a period, **ddname**, it indicates that there is a ddrec defined to pcMAINFRAME that will be used to dynamically allocate the file. If the ddname is not preceded by a period.
pcMAINFRAME expects the ddname to be allocated prior to the file transfer. If a PDS is pre-allocated, the member name must be specified.

**File Type:** this **required** entry will be: **ESDS, RRDS, KSDS, LIB, PUBL, EXIT, USER, PDS or SEQ**, indicating the type of file named in the previous entry. **ESDS, RRDS, and KSDS** describe VSAM files. **EXIT** is used to indicate that the previous entry is the PPT name of a user exit program. **LIB and PUBL** indicate that this profile is using the pcMAINFRAME Private (LIB) or Public (PUBL) library facility. **PDS** is used to indicate a partitioned data set and **SEQ** refers to any standard QSAM data set. **USER** is used to define user profiles and is explained in the User Profile Maintenance section on page of this manual.

For LIB and PUBL uploads, the File Name specified will be overridden by the library file names specified in the user’s ID record **unless** the user’s ID record indicates "*" in those fields. For LIB and PUBL downloads, the File Name specified in the profile is always used. See Appendix I for a description of the differences between libraries and public libraries.

**CICS:** valid entries are ESDS, RRDS, KSDS, LIB, PUBL, EXIT, and USER.

**TSO:** valid entries are ESDS, KSDS, LIB, PUBL, PDS, and SEQ.

**Output Data Format:** this must be specified as: **A - ASCII, B - BASIC, D - DIF, F - fixed, I - Include, L - Excel, T - transparent, S - spool, V - Btrieve, W - WKS, or X - xBASE** to indicate the output format of the data that is being transmitted. This is a **required** field. Descriptions of these formats follow:

**ASCII / Text** a data format that produces files with trailing spaces removed from each record. In other respects it is similar to the **FIXED** format below.

**BASIC** a data format commonly used by programs written in the programming language BASIC and sometimes known as "comma-delimited". Character fields are prefixed and suffixed by quotation marks ("), fields are separated by commas (,), character fields are shortened by removing trailing blanks and numeric fields contain optionally a leading sign, decimal point, and are shortened by removing any leading zeros. Records are delimited by a carriage return and line feed character pair (CRLF). Records and fields are variable in length.

**Btrieve** this supports the data format used for Btrieve files. See **Appendix A** for full details.

**Excel** the worksheet format used in the Microsoft Excel spreadsheet program.

**DIF** Data Interchange Format, a standard data format used by many microcomputer applications, most notably the various spreadsheet programs, for transferring information.

**Fixed** This format is most frequently found in PC ASCII files such as text files (for word processing) or data files (for COBOL compilers). A fixed file is delimited by CRLFs. Its records and fields are in fixed positions but records may be of varying lengths. Although most text
files do have both a CR (carriage return) and a LF (line feed) at the end of each record, records delimited with just a CR or LF can be uploaded by pcMAINFRAME. During downloads, CRLF delimiters are written.

Fixed length records that are not CRLF delimited are also supported by pcMAINFRAME. Use the fixed record length field in the profile to define record size. When a fixed record length is specified, a CRLF will not be written during download. Any CRLF found in the file will be treated as data unless it falls at the end of a record. Do not specify a fixed record length if your file is delimited by CRLF.

**Include** indicates that several fixed files are to be combined to produce a single file. It is useful, for example, to insert a data file into a file containing JCL prior to uploading to a READER QUEUE. This data type is only valid for uploads.

**Transparent (Binary)** transfers all binary values between the PC and the mainframe. This may be used to upload or download: programs in object form, spreadsheet models in binary format, and other non-text files. Data is not reformatted in this mode, so a file transferred from the PC to the mainframe would appear on the mainframe exactly as it appears on the PC. Any text in the file would still be in ASCII format and control characters would not be in their normal mainframe representation.

**Spool** transfers print data between the host spooling system and the PC. Form feed characters and spacing commands are translated from mainframe format to those used on PC printers.

**WKS** worksheet format as used by many popular spreadsheets such as LOTUS, Quattro, WK1 and WKS, all of which are supported.

**xBASE** the database format used by many PC databases such as dBASE, FOXbase, and Clipper. dBASE III format is used unless the number of fields exceeds 128 in which case dBASE IV format is used. For uploads, either dBASE III or IV is acceptable.

**Data Transfer Direction**: this **required** field indicates, with a **U**, **D** or **B**, the transfer direction(s) this profile allows. Specifying a **D** means download only, **U** means upload only, and **B** means both upload and download.

**A note about uploads**: During an upload, re-formatting takes place in the opposite way it does during a download. ASCII data on the PC is translated to EBCDIC, numeric data fields are converted to zoned decimal, binary or packed as specified in the profile, and DIF, BASIC, Fixed, or Spool file types on the PC are converted to the mainframe format.

**No selection or browsing takes place during an upload**: all records in the PC file are uploaded to the mainframe. The upload may go to any supported file type. All records are added to the selected mainframe file except KSDS type files. If the record exists on the mainframe it will be overwritten with PC data; otherwise, PC data is added to the KSDS file. If uploads are done to KSDS files, the full key field of the file must be one of the data fields uploaded.

**Generate Column Headers**: this field is used for WKS, Excel, BASIC and DIF files and is not appropriate for other types. If **Y** is selected, column heading labels derived from the field names will be generated. These
will appear on spreadsheets into which the downloaded data is subsequently loaded.

**Read Limit:** this **required** field must contain a number between 0 and 999,999, indicating the number of mainframe records to be read during a request before terminating it. This facility is a "safety valve" that prevents an undetected parameter entry error from causing excessive file accesses. When the limit is reached, the request is terminated and control is returned to the menu. Entry of a non-zero value will cause the request to terminate when that number of input records is reached. A **zero value entry will disable the check.**

**Write Limit:** this entry functions the same as Read Limit but controls the number of records that are written to the Mainframe. When the number of records sent from the PC exceeds the limit, the request is terminated. **If the value is set to zero the function is disabled.**

**Skip type:** this required field must contain a **B** to indicate skip before or an **I** to indicate selecting every n\textsuperscript{th} record. The number of records skipped before downloading with a type B is indicated in the count value. If type I is specified, the last record in each count interval will be selected. For example, if count is 50 every 50\textsuperscript{th} record will be downloaded.

**Count:** value between 0 and 99,999. If the value is entered as zero no skipping occurs.

**Fixed Record Length** this optional field applies only to profiles of Fixed or Transparent data types. It defines the size of the record in cases where it cannot be determined from the file, or when you wish to over-ride the record size. It is also useful for creating PC files for applications that do not have CRLF record delimiters. Fixed type records on the PC are normally terminated with a CRLF combination. If the above parameter is specified, no CRLF will be written on downloads and none will be required on uploads. Records of the size specified will be transferred. On upload, the last record will be padded with spaces or low-values if required. **Note:** a CRLF is treated as data unless it occurs at the end of the defined logical record length.

Transparent format files are read and written with the specified size records.

**Download Count Limit:** Defined only when the file type is defined as LIB or PUBL. This establishes the maximum number of times a member may be downloaded. For example, this is used to make sure data is not inadvertently processed more than once. The count may be updated through the **Library Directory** screen.

**Timestamp Control.** Defined only when the file type is defined as LIB or PUBL. This allows you to timestamp the file created by the profile, preventing users from overwriting PC files more current than the library file, and from overwriting library files more current than a PC file.

Select **Y** to timestamp library members when new members are created or when existing members are updated.

Select **M** to mark library members when new members are created or when existing members are updated.

Select **N** to eliminate timestamping for library members when members are created or updated. This selection has no impact on downloads.
Translation Modes: This field must be U, M, or W to indicate Upper case, Mixed case, or Word processing mode. Upper case will cause all lower case text to be translated to upper case before transfer, while Mixed case will allow both upper and lower case text to be transferred. Word processing mode transfers upper and lower case text, and permits non-text data (for example, embedded control characters) to be transferred with no translation. The non-text characters are transferred as hexadecimal values without translation. This means that a hex '01' on the PC will remain a hex '01' on the mainframe since it is not a text character as defined in Appendix B - Valid Text Characters of this manual. An editor on the mainframe can then replace the hex '01' with the mainframe control character of the same function. Some discretion must be used in selecting the control characters used in this type of text. Certain values of hex on the PC may become valid characters once the file is transferred to the mainframe. For example, the hex value 'C1' is not a valid character on the PC but when transferred to the mainframe represents the letter "A". If control characters are limited to those values below hex '7A' these problems will generally be avoided.

Write Ctrl_Z at EOF: Enter either Y or N to indicate whether a \t (hex 1A) should be written as the last character on the PC file being downloaded. This field has no effect on uploads or downloads with a Transparent data format.

Make blank fields Null: Enter either Y or N to indicate whether during a DIF or Basic download, blank fields should be shipped as a null field (i.e. ""). This field has no effect on any other data type. This field also has no effect during uploads.

Multiple Record Header (Y/N): This field, when "Y", indicates that the file to be downloaded has multiple record formats. A subsequent screen will be used to define the specific records types and pointers to the profiles that process each record type.

Last Maint/Used Dates: These display the date that this profile was added or last changed and the date that the profile was last used in a transfer.

Permit Upload Overlay: Used for LIB or PUBL type profiles only, this controls whether a second upload of a member will overwrite the first upload. If the switch is set to N, the second upload is not permitted. The switch may be updated with the Library Directory screen or a batch program that processes the member.

Download History Log: Used for LIB or PUBL type profiles only, this controls log designation. If the switch is set to Y, activity of a file created or updated with this profile will be logged. If the switch is set to N, no activity of a file created or updated with this file will be logged.

The pcMAINFRAME system edits all input and/or changes made on the screen and displays error messages if necessary. When all errors are corrected, a prompt for more input or an update indicator is displayed. Pressing \t will advance the display to the next screen. If input or change is complete, the appropriate update code should be entered to cause an add, delete, or update to take place. (Return to this screen after all input or changes are complete on subsequent screens to perform the actual update.)
Field Description Screens

The rest of the screens in the profile maintenance set are classified as “field description” screens. They are used to define exactly what data is to be transferred, and to and from which format. The seven different field description screens are titled: Access Ranges and Limits, Record Selection, User Exit Parameters, Special Exit Parameters (used for Btrieve processing), Record Type Definition, Break Control and Output Format.

These screens also have common control characteristics. Each screen supports multiple "field description" lines and provides a common method of data entry. Control character(s) may be placed at the beginning of any detail line to perform insertion, duplication, and deletion functions.

- To duplicate a detail line place an equal sign, = in position one of the data line to be duplicated and press \[Enter\]. The line marked with the = is duplicated and lines below are all moved down one line.
- Similarly a > or a >A placed at the beginning of a detail line will insert a blank line following the marked line.
- A < will delete the marked line and cause all subsequent lines to move up one position.
- A >B will insert a blank line before the marked line.

Multiple control characters may be placed on the screen at one time if several functions are required. Not all of the "field description" screens are presented for every profile. Options specified in the profile, such as transfer direction and file type, determine which screens will be used.

Variable Parameters

An extremely powerful feature of pcMAINFRAME is its ability to have users input the values for selection, browsing, and insertion dynamically at file transfer time. Variable parameters can be used in all of the "field description" screens. You can code a variable in place of any "comparison value" or literal value in the field description screens. The PC user will be prompted to enter the appropriate value for the variable when the file transfer is executed.

A variable parameter is specified by making the first character of the field an ampersand (&). All characters following the ampersand will be used as the prompt for the PC.

An example of the use of variable parameters is given in Figure 4-5. In this case, the PC will be prompted to "ENTER FROM DATE" and "ENTER TO DATE" when the profile "SAMPLE" is invoked. The PC user may either enter the desired value or a ./. (period-slash-period) to bypass that selection criteria or enter a null data value.

If you have a transfer running in unattended or automated mode, the system will get the value for your variables from a "setup file" on the PC.
### Access Ranges and Limits Screen

If the file type is KSDS, the Access Ranges and Limits screen shown in Figure 4-6 is displayed.

<table>
<thead>
<tr>
<th>Profile: SAMPLE</th>
<th>Profile Definition</th>
<th>Page 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Ranges and Limits</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data/Field Name:</th>
<th>Browse Range: From (&gt;=)</th>
<th>To (&lt;=)</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGION1</td>
<td>01</td>
<td>01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Commands:</th>
<th>&gt;A - Insert after, &gt;B - Insert before, = - Duplicate, &lt; - Delete.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF1 - 1st page, PF2 - Next Page, PF3 - Prior Page, CLEAR - Menu.</td>
<td></td>
</tr>
</tbody>
</table>

This screen is optional and provides keyed files with a facility for limiting access to certain portions of the file. Up to ten low - high ranges of keys or partial keys may be specified. When a request that uses this file is processed, only records within the ranges specified will be read and evaluated for downloading to the PC. This facility can save a great deal of time when processing large files. The data required on this screen is as follows:

- **Data/Field Name:** this is optional and is used only for documentation purposes. One to twelve characters may be input to describe or name the range.

- **From:** this is the low or beginning value of the range. This field may contain from one to twenty five characters of data or a variable parameter. The data entered is treated as a partial key and padded with low-values. Records within the file with keys or partial high-order keys equal to or greater than the value specified are selected for further evaluation.

- **To:** this is the high or ending value of the range. The one to twenty five characters of data entered is treated as a partial key and padded with high-values. Record keys are compared to this value and once a greater than condition is found, processing is terminated or the next range is processed if present. The To value need not be of the same length as the From value.

The high value of each range must be equal to or greater than the low value for that range, and the low value for every range must be greater than the preceding high value. Non-character values in the keys may be entered in hexadecimal format by
bracketing the values with ( ) parentheses. Values entered in hex must be an even number of characters and may consist of pairs of valid hex characters (0-9, A-F). If a field is entered in hex in one field, the corresponding position in the other field must also be entered in hex. For example, if the first field low range is ABCD(F109), the 5th and 6th position of the high range (if used) must also be in hex. EFGH, EFGH(F3) and EFGH(F304) would be valid but EFGH3 would not be valid. If no browse ranges are entered, the entire file will be processed.

**User Exit Parameters Screen**

If the profile specifies an EXIT, *User Exit Parameter* screen as seen in Figure 4-7 is displayed.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Maximum Length</th>
<th>Data Type</th>
<th>Dec Plc</th>
<th>Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Length</td>
<td>Type</td>
<td>Plc</td>
<td>Value</td>
</tr>
</tbody>
</table>

Define user exit parameters...
from none to 12 are allowed.

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete. PF1 - 1st page, PF2 - Next Page, PF3 - Prior Page, CLEAR - Menu.

**Figure 4-7: User Exit Parameters**

From zero to twelve parameters may be entered which will be made available to the user exit when it executes. The following fields may be entered for each parameter:

**Parameter Name:** this is an optional one to twelve character description of the parameter.

**Maximum Length:** this required field is the length of the parameter. The maximum value is 25 bytes for character fields and 18 bytes for numeric fields.

**Data Type:** this required field indicates the type of data contained in the parameter. Acceptable values are C - character and N - numeric.

**Dec Plc:** this field is used to indicate decimal places in any numeric parameters. The number of decimal places may be entered as an integer between one and the length of the parameter. If no entry is made, zero decimal places is assumed.

**Parameter Value:** this required field may contain either the value for the parameter up to the length specified or a variable parameter prompt. Variable parameters were discussed in detail earlier in this chapter. Appendix C - Coding User Exits covers the writing and uses of user exits.
Multiple File Definition Screen

The Multiple File Definition screen shown in Figure 4-8 is displayed for all types of downloads. It is not displayed when a Multiple Record Header is defined.

```
<table>
<thead>
<tr>
<th>File</th>
<th>Field</th>
<th>Type</th>
<th>Rule</th>
<th>Name</th>
<th>Pos</th>
<th>Len</th>
<th>P</th>
<th>D Pos</th>
<th>Len</th>
<th>P</th>
<th>Literal Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>ITEMMST</td>
<td>KSDS</td>
<td>C</td>
<td>ITEMNO</td>
<td>1</td>
<td>4</td>
<td>N</td>
<td>*</td>
<td>25</td>
<td>7</td>
<td>C</td>
</tr>
<tr>
<td>B</td>
<td>VENDMST</td>
<td>K-GE</td>
<td>T</td>
<td>VENDOR</td>
<td>1</td>
<td>9</td>
<td>C</td>
<td>A</td>
<td>33</td>
<td>9</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>TAXMAST</td>
<td>KSDS</td>
<td>A</td>
<td>STATE</td>
<td>1</td>
<td>2</td>
<td>C</td>
<td>B</td>
<td>59</td>
<td>2</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>TAXMAST</td>
<td>KSDS</td>
<td>T</td>
<td>STATE</td>
<td>1</td>
<td>2</td>
<td>C</td>
<td>B</td>
<td>59</td>
<td>2</td>
<td>C</td>
</tr>
<tr>
<td>@</td>
<td>TAXMAST</td>
<td>KSDS</td>
<td>T</td>
<td>STATE</td>
<td>3</td>
<td>5</td>
<td>C</td>
<td>A</td>
<td>82</td>
<td>5</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>TAXMAST</td>
<td>KSDS</td>
<td>ALL_ZERO</td>
<td></td>
<td>3</td>
<td>5</td>
<td>C</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>L   =0</td>
</tr>
</tbody>
</table>
```

This screen allows the user to define up to six ‘secondary’ files to be accessed during data extraction. The files must be VSAM KSDS, and a full key or partial (generic) key to the file must be available. Key data can be obtained from the primary record being downloaded and from any previously defined secondary file.

For example, as shown in figure 4-8: during the download of an inventory file (the primary file), an item number, ITEMNO (from the primary record), can be used to retrieve the corresponding item master record in the ITEMMST file. The vendor master record in the VENDMST file is retrieved using the VENDOR field from the ITEMMST as the key. The tax rate is determined by retrieving the record from the TAXMST file with a key built from the STATE field of the VENDMST file and the TAXCAT from the ITEMMST file.

Below, Figure 4-9 shows the illustrates a Keyed Browse (K-BR) request. Again, the inventory master is the primary file, and each ITEM and VENDOR is accessed directly. The Tax Master (TAXMAST) file, however, is to be accessed with a ‘generic browse’. This download will result in a record for each tax category for a given item/vendor on the inventory file.
Each entry on this screen defines:

- IDENTIFICATION of the file to be accessed
- KEY FIELD(s) used to access the file
- SOURCE FIELDS where key field data can be found

**IDENTIFICATION Fields**

**ID:** This is a unique 1-character identifier assigned to the file.

- Entries from A-Z are allowed. This USER assigned ID is used in subsequent screens to indicate that the field being defined is in the secondary file with the given ID. Once an ID is assigned to a file, it will not be changed by the system.
- @ This identifier is used in conjunction with the NOT FOUND rule of Alternate record. It is used immediately following the file designated as the alternate key. *(See Not Found rule)*

**File Name:** This is the FCT name of the file to be accessed. For TSO users, this is a DDNAME. If a DDREC reference is used (.DDRECID), the file will be dynamically allocated during transfer. Otherwise the file must be pre-allocated to the TSO session with the specified DDNAME.

A file name can be associated with only one file ID.

**File Type:** There are three file types allowed for use with Multiple File definition: KSDS, K-GE and K-BR.

- **KSDS** - this indicates that the file type is a keyed sequence data set, using the full key of the file. This should be used unless one of the following two special circumstances is required.
- **K-GE - Keyed Generic.** Access a secondary file using a generic key. This partial key must start in the first position of the full key. The read will obtain the first record on the file that has the same starting characters as the data specified. One and only one record from the secondary file will be obtained for each record selected from the primary.
- **K-BR - Keyed Browse.** This denotes a KSDS file, using a generic key to access one or more secondary records sequentially. The request will obtain each record on the secondary file that has the same starting characters as the data specified by the generic key. A record will be extracted and downloaded for EACH matched record on the secondary file, as long as select-include/exclude rules are true. The request is similar to an SQL "join" request.

**Not Found Rule:** If a record in a secondary file cannot be found specify one of the following actions:

- **A** - continue processing by looking at the alternate key. The alternate definition follows this entry, and is designated by an @ in the file-id.
- **T** - terminate the transfer. This is the default action.
- **C** - continue processing as if the data were spaces in a character field or zero in a numeric field. If any other secondary files specify
selection criteria from the not found secondary, the record is not selected.

- S - skip the record being formatted is skipped and nothing for the record is downloaded.

**KEY Fields**

**Field Name:** This field is the user defined field name.

**Begin Pos:** This *required* field defines the location of the key field in the file to be accessed.

**Fld Length:** This *required* entry defines the length of the KEY field. The maximum length of a field depends on its data type:

- Binary = 2 4 or 8 bytes
- Character = up to 128 bytes
- Numerics = up to 18 bytes

**TP:** This *required* entry defines the key fields data type. Valid types are:

- B (Binary)
- C (Character)
- N (unsigned numeric)
- P (Packed)
- S (signed numeric)

Keys may be built from multiple fields. Define one line for each segment of the key as necessary. Figure 4-8 shows an example of this.

**SOURCE Fields**

**ID:** This *required* field specifies the ID of the file that must be used to find the key data. An ‘*’ indicates that the key data is in the primary record, an ‘A’ indicates that the data for the key can be found in secondary file ‘A’, etc.

Data for a key field must come from the primary file or a previously defined secondary file.

**Begin Pos:** This *required* field defines the location of the source data for the key. It can be any location in the specified source file.

**Fld Length:** This *required* entry defines the length of the source data field. The maximum length of a field depends upon its data type:

- Binary = 2 4 or 8 bytes
- Character = up to 128 bytes
- Literals = up to 15 bytes
- Numerics = up to 18 bytes
**TP**: This *required* entry defines the source field’s data type. Valid types are:
- B (Binary)
- C (Character)
- L (Literal)
- N (unsigned numeric)
- P (Packed)
- S (signed numeric)

If the KEY field and the SOURCE field are numeric with different data types, the data will be converted to the key field data type prior to the file access.

**Literal Data**: This entry is used only when the source data type is Literal. The values coded in this field (up to 15 bytes) will be moved to the KEY field before the read.

The following figurative constants are also valid:
- =SPACES
- =ZEROS
- =LOW-VALUES
- =HIGH-VALUES
- =X , where X is any fill character

**Record Selection Screen**

The *Record Selection* screen shown in Figure 4-10 is displayed for all types of downloads.

![Record Selection Screen](image)

This screen allows users to enter zero to twelve selection criteria which control inclusion and exclusion of records to be downloaded to the PC. The elements on the record selection screen identify field criteria that determine if a particular record should be transferred. *If no selection criteria are specified, all records will be selected.*

A test may be established to stop further processing of the file upon a condition being satisfied. This reduces the number of records read on the mainframe.
Up to 16 conditions or eight ranges may be specified to determine records to be processed. Selection criteria may be logically "ANDED" or "ORed" together. Selection elements are as follows:

**Data/Field Name:** this field is *required* and may contain from one to twelve characters. Its text will be used in error messages and should be descriptive of the field it identifies.

**Begin Pos:** this is the location in the mainframe record at which the field being described begins. This must be within the maximum record length pcMAINFRAME can work with, 32,767 bytes. This field is *required*. The value is relative to one so that the first byte of the record is position one, not zero.

**Bytes:** this is the length of the field in the input record. This entry is *required* and must be less than or equal to the maximum comparison value length of 25 bytes (18 bytes if the field is numeric).

**Data Type:** this *required* field specifies the format of the data contained in this field. Valid entries are: B - binary, C - character, D - date (mmddyy or mm/dd/yy), I - International Date (ddmmyy or dd/mm/yy), L - long date format (mmddccyy or mm/dd/ccyy), M - long international format (ddmcccyy or dd/mm/ccyy), N - numeric (zoned decimal), or P - packed. It is very important to specify data type correctly so that the selection comparisons will be accurate. *Note:* L and M data types can have a field length or either 8 or 10 bytes.

**Dec Plc:** this entry is invalid for non-numeric entries, types C and D. The number of decimal places may range from 0 to 18. If no entry is made, 0 is assumed.

**Action:** this entry must be an I, E, S, W, or A (which is then changed to &) to indicate: Inclusion or Exclusion, Stop-after, start-When, or And respectively. The rules below are listed in order of their actual priority:

1. The W or start when, is used to specify a condition that triggers the start of transfer. If start when is used, all records will be bypassed until a record satisfying the start when condition is encountered. All subsequent records in the file will be evaluated according to any include and exclude criteria.
2. The S or stop after is used to indicate a condition which, if satisfied, will cause processing of a file to terminate. This can reduce the amount of data read at the mainframe.
3. Records matching the criteria for any exclusion entry are rejected immediately.
4. Records matching the criteria of any inclusion entry are selected immediately.
5. If no inclusion criteria are specified, all records not excluded are included.
6. An ampersand character, &, may be used to "and" two or more conditions together into a single test. The conditions to be "anded" together must be consecutive lines on the screen with all but the first having & as the action.
If: this required field specifies the logical operator to be used in comparing the contents of the input record field and the comparison value in the next field. Valid entries for this field are: =, <>, <, >, >=, <=, LO, and HI. LO and HI must be specified together on adjacent lines and specify an inclusive range for selection. Figure 4-5 shows an example of a LO-HI range being used for date selection.

Comparison Value: this required entry contains from 1 to 25 characters of data or a variable parameter. The data entered is tested against the input record during transmission to determine if the record should be transferred. Leading signs and decimal points should be entered if required on numeric fields. Notes, messages and comments are added to this field by pcMAINFRAME.

Profile List - Record Type Definition

If the profile is defined as a "Multiple Record Header", the Profile List / Record Type Definition screen as shown in Figure 4-11 is displayed.

<table>
<thead>
<tr>
<th>T</th>
<th>Profile Parent</th>
<th>Fld</th>
<th>Fld</th>
<th>Data</th>
<th>Dec</th>
<th>Action</th>
<th>Comparison Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>PCMTMRH1</td>
<td>1</td>
<td>2</td>
<td>C</td>
<td>I</td>
<td>=</td>
<td>01</td>
</tr>
<tr>
<td>B</td>
<td>PCMTMRH2</td>
<td>1</td>
<td>2</td>
<td>C</td>
<td>I</td>
<td>=</td>
<td>02</td>
</tr>
<tr>
<td>C</td>
<td>PCMTMRH3</td>
<td>0</td>
<td>0</td>
<td>C</td>
<td>I</td>
<td>OT</td>
<td>(All other records)</td>
</tr>
</tbody>
</table>

Define Profiles and Record ID's...
From 1 to 12 are allowed.

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete.
PF1 - 1st page, PF2 - Next Page, PF3 - Prior Page, Clear - Menu.

Figure 4-11: Profile List - Record Type Definition

This screen allows users to define up to twelve different record types in the same download. It is used to identify each record type on the file, and to point to individual profiles that contain specific select and output format definitions for each record type.

One or more lines can be used to define a single record type. In the example, record type "A" will be processed by profile PCMTMRH1 and is identified by a "01" in position 1 of the record AND a date greater than or equal to 01/01/96 Record type "B" will be processed by profile PCMTMRH2 and is identified by a "02" in position 1 of the record. Any other record type will be processed by profile PCMTMRH3.
Multiple Record Hierarchy

There is support for defining a hierarchy of records and ensuring that child records will not be selected without their parent’s selection. Further you may specify that parent records will not be selected without at least one child record. The figure below demonstrates the use of those functions.

<table>
<thead>
<tr>
<th>Profile List / Record Type Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
</tr>
<tr>
<td>P</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>E</td>
</tr>
<tr>
<td>F</td>
</tr>
<tr>
<td>G</td>
</tr>
<tr>
<td>H</td>
</tr>
<tr>
<td>I</td>
</tr>
</tbody>
</table>

Define Profiles and Record ID's...

From 1 to 12 are allowed.

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete.
PF1 - 1st page, PF2 - Next Page, PF3 - Prior Page, Clear - Menu.

Figure 4-12: Multiple Record with Hierarchy Definition

There are some limitations to MRH processing, as follows:

- Uploads are not supported;
- Only Fixed and Basic output formats are supported;
- Break processing (summarization, average, etc.) is not supported;
- Browse ranges on individual profile records are ignored.

The following defines the coding rules for the Profile List Screen.

TP (Record Type): This field is not enterable and is used to collect all of the field qualifications for a single type of record. It is valued by the system when the profile name field changes.

Profile Name: This field is **required** and contains the name of the profile that will be used to download the specified record type. This field is NOT validated at profile definition time. If a download is run and the profile specified does not exist, the error **PCM004 Profile Not Found** will be issued.

Parent ID: This field is **Optional**. The record type of this field’s immediate parent in a hierarchy. When a record has a Parent ID, then that record will not be selected for processing if its parent is not selected for processing.

Child Required?: This field is **Optional**. This field indicates that a child record must be selected before the parent record can be written. This field prevents the selection and transfer of parent records without children.
**ID Field Position:** This field is *required* and defines the beginning location in the mainframe record of a field that identifies the record type. This must be within the first 4096 bytes of the record. When defining a "default" profile used to process "all other" records, this field should be left blank.

**Field Length:** this is the length of the field in the input record. This entry is *required* and must be less than or equal to the maximum comparison value length of 25 bytes (18 bytes if the field is numeric).

**Data Type:** this *required* field specifies the format of the data contained in this field. Valid entries are: **B** - binary, **C** - character, **D** - date (mmddyy or mm/dd/yy), **I** - International Date (dmmmyy or dd/mm/yy), **L** - long date format (mmddccyy or mm/dd/ccyy), **M** - long international format (dmmccyy or dd/mm/ccyy), **N** - numeric (zoned decimal), or **P** - packed.

**Action:** The action to take when the record matches the select criteria. For record-type definitions, this field must be an "I" (include) or an "&" (include if this condition AND the prior condition). Records matching the criteria for inclusion are selected and handed to the profile specified in the profile name field for final selection, formatting and download.

**If:** this *required* field specifies the logical operator to be used in comparing the contents of the input record field and the comparison value in the next field. Valid entries for this field are: **=**, **<>**, **<**, **>**, **>=**, **<=**, **LO**, **HI** and **OT**. **LO** and **HI** must be specified together on adjacent lines and specify an inclusive range for selection. The value **OT** can be used to specify a default profile. If a record does not match the criteria specified for any other profile and **OT** is defined, then the named profile will be used. If **OT** is not defined, then the record will not be selected.

**Comparison Value:** this *required* entry contains from 1 to 25 characters of data or a variable parameter. The data entered is tested against the input record during transmission to determine the record type. Leading signs and decimal points should be entered if required on numeric fields. Notes, messages and comments are added to this field by pcMAINFRAME.
Break Control Screen

The Break Control screen as shown in figure 4-13 is displayed for all types of downloads. The mechanics of this screen are the same as those of the previous screens. Use PF2, PF3, and PF1 to advance to the next screen, go back to the previous screen, and go to the first screen respectively.

<table>
<thead>
<tr>
<th>Data/Field Name</th>
<th>Begin Pos</th>
<th>Bytes</th>
<th>Data Type</th>
<th>Decimal Places</th>
<th>Break If</th>
<th>Comparison Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGION</td>
<td>1</td>
<td>2</td>
<td>N</td>
<td>ANY</td>
<td>ANY</td>
<td>(Any break in field)</td>
</tr>
<tr>
<td>DISTRICT</td>
<td>3</td>
<td>2</td>
<td>N</td>
<td>ANY</td>
<td>ANY</td>
<td>(Any break in field)</td>
</tr>
<tr>
<td>BRANCH</td>
<td>5</td>
<td>4</td>
<td>N</td>
<td>ANY</td>
<td>ANY</td>
<td>(Any break in field)</td>
</tr>
<tr>
<td>PRODUCT</td>
<td>9</td>
<td>6</td>
<td>C</td>
<td>ANY</td>
<td>ANY</td>
<td>(Any break in field)</td>
</tr>
</tbody>
</table>

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete.
PF1 - 1st page, PF2 - Next Page, PF3 - Prior Page, CLEAR - Menu.

This optional screen is used to define the points at which group or summarization functions are "broken" and a record containing the accumulated values is to be generated. From one to six fields may be specified that may either collectively or independently trigger a break. Note: this is a single level break and the specification of multiple fields is a way of combining several fields from the input record into a larger control field, **but it does not create multiple levels of break**. If this screen is used, an output record is created every time a break occurs. If this screen is not used, an output record is created for every input record that passes the browse and record select criteria.

The data elements are:

- **Data/Field Name**: this field is required and may contain from one to twelve characters. Its text will be used in error messages and should be descriptive of the field it identifies.

- **Begin Pos**: this is the location in the mainframe record, relative to one, at which the field being described begins. This must be within the maximum record length of 32,761 bytes that pcMAINFRAME can work with. This field is required.

- **Bytes**: this is the length of the field in the input record. This field is required and must be less than or equal to the maximum comparison value length of 25 bytes (18 bytes if the field is numeric).

- **Data Type**: this required field specifies the format of the data contained in this field. Valid entries are: B - binary, C - character, D - date (mmddyy or mm/dd/yy), L - long date format (mmddccyy or mm/dd/ccyy), M - long international format (ddmmccyy or dd/mm/ccyy), N - numeric (zoned decimal), or P - packed. It is very important to specify data type correctly so that the selection comparisons will be accurate.

- **Dec Plc**: this entry is invalid for non-numeric entries, types C and D. The number of decimal places may range from 0 to 18. If no entry is made, 0 is assumed.
Break If: this required field specifies the logical operator to be used in comparing the contents of the input record field and the comparison value in the next field. Valid entries for this field are: =, <>, <, >, >=, <=, and ANY. ANY causes a break to occur if there is any change in the input field.

Comparison Value: this optional entry contains the character representation of the value in the input record to be tested. If the Break If field specified ANY, no entry is required in Comparison Value. Leading signs and decimal points should be entered if required on numeric fields. This field may also be entered as a variable parameter. Notes, messages and comments are added to this field by pcMAINFRAME.

Output Format Screen

The final field description screen is the Output Format screen as shown in Figure 4-14. This screen is needed for all profiles except for spool and transparent transfers. The same mechanics for paging forward, paging backward and paging to the first screen apply to the Output Format screen as those that were discussed for other screens above.

The output format screen specifies the content and location of fields in the mainframe file that are to be transferred. It also specifies the lengths and data types of PC record fields. Each line on the screen represents a field that is to be output to the PC. The order in which the fields are placed on the screen is the order in which they will appear in the output record sent to the PC during download and the assumed order of the fields in the PC record during upload.

During an upload operation, reverse field formatting takes place. The PC fields in the order specified in the Output Format screen are reformatted and placed in mainframe record at the location defined in Begin Pos. To transfer the entire record, only the field name and operator "ALL" are required.

You may specify up to 500 fields on multiple pages of the Output Format screen. The line commands used to "insert", "delete" and "duplicate" discussed previously are particularly helpful in rearranging fields and building the output record.

Tables of fields (fields that repeat n number of times) can be defined. When defining a table, two additional lines, Fixtable/Vartable and Endtable, are needed to mark the

<table>
<thead>
<tr>
<th>Data/Field Name</th>
<th>Begin Pos</th>
<th>Size Type</th>
<th>Pic</th>
<th>Operation Picture</th>
<th>Inserted Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGION</td>
<td>1</td>
<td>2</td>
<td>N</td>
<td>MOVE</td>
<td>Numeric S9(2)</td>
</tr>
<tr>
<td>BRANCH</td>
<td>5</td>
<td>4</td>
<td>N</td>
<td>MOVE</td>
<td>Numeric S9(4)</td>
</tr>
<tr>
<td>PROD TABLE</td>
<td>9</td>
<td>10</td>
<td>O</td>
<td>FIXTABLE</td>
<td>Fixed Occurs 10 TIMES</td>
</tr>
<tr>
<td>PRODUCT</td>
<td>1</td>
<td>6</td>
<td>C</td>
<td>MOVE</td>
<td>Alpha/Num X(6)</td>
</tr>
<tr>
<td>DOLLAR SALES</td>
<td>7</td>
<td>6</td>
<td>P</td>
<td>MOVE</td>
<td>Packed S9(9)V9(2)</td>
</tr>
<tr>
<td>UNIT SALES</td>
<td>13</td>
<td>3</td>
<td>F</td>
<td>MOVE</td>
<td>Packed S9(5)</td>
</tr>
<tr>
<td>BILLING CODE</td>
<td>16</td>
<td>1</td>
<td>C</td>
<td>MOVE</td>
<td>Alpha/Num X(1)</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td>ENDTABLE</td>
<td></td>
</tr>
<tr>
<td>DESC TABLE</td>
<td>169</td>
<td>32</td>
<td>O</td>
<td>VARTABLE</td>
<td>Var Occurs 0 to 32 TIMES</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>MOVE</td>
<td>Alpha/Num X(1)</td>
</tr>
<tr>
<td>170</td>
<td></td>
<td></td>
<td></td>
<td>ENDTABLE</td>
<td></td>
</tr>
</tbody>
</table>

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete.

PF1 - 1st page, PF2 - Next Page, PF3 - Prior Page, CLEAR - Menu.
beginning and end of the table. During transmission, pcMAINFRAME expands the table definition by repeating each field and processing the expanded definition as a normal profile. In this case, the expanded profile cannot exceed the 500 fields limit.

Only one variable table can be defined per profile, and it must occur at the end of the record. When transferring a record using a profile that has a variable table defined, the data record must be at least as long as the fixed portion of the definition.

The data elements used by the Output Format screen are as follows:

**Data/Field Name:** this field is **required** and may contain from one to twelve characters. Its text will be used in error messages and should be descriptive of the field it identifies. If you are doing an xBASE download, this field cannot be larger than ten characters and will be used as a field name in your xBASE file.

If the field is within a table, the last three characters will be overlaid with the occurrence number. The names given to a table field will consist of the original field name plus a three digit sequence number. If the field name is longer than seven characters, only the first seven will be used because xBASE files only support ten character field names.

Dashes (-) will be replaced with underscores ( _ ) in xBASE transfers because dashes are invalid in xBASE field names.

**Begin Pos:** this is the location in the mainframe record, relative to one, at which the field begins. This field is **required**.

**MF Size:** this is the length of the field in the mainframe record. This field is **required**.

**MF Type:** this **required** field specifies the format of the data contained in this field. It is very important to specify the data type correctly so that the data conversion will be accurate. The valid entries are:

**A:** Binary Unsigned. 1, 2, 4, or 8 byte binary fields. These are unsigned values in the range of:

<table>
<thead>
<tr>
<th>Length</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 - 255</td>
</tr>
<tr>
<td>2</td>
<td>0 - 65,535</td>
</tr>
<tr>
<td>4</td>
<td>0 - 4,294,967,295</td>
</tr>
<tr>
<td>8</td>
<td>0 - 999,999,999,999,999</td>
</tr>
</tbody>
</table>

**B:** Binary. 1, 2, 4, or 8 byte binary fields are valid. These are standard signed binary fields. The range of values in the various length fields is:

<table>
<thead>
<tr>
<th>Length</th>
<th>Low Limit</th>
<th>High Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-128</td>
<td>+127</td>
</tr>
<tr>
<td>2</td>
<td>-32768</td>
<td>+32767</td>
</tr>
<tr>
<td>4</td>
<td>-2,147,483,648</td>
<td>+2,147,483,647</td>
</tr>
<tr>
<td>8</td>
<td>-999,999,999,999,999</td>
<td>+999,999,999,999,999</td>
</tr>
</tbody>
</table>

**C:** Character. Standard text format in EBCDIC on the mainframe and ASCII on the PC.
**E:** Edited Numeric. This converts a field that is in display format as in a printed report ($53,321.05CR) and converts it into a standard number. This may be used to convert edited numeric printouts back into numeric data types. Associated with the edited numeric data type is a logical data type. This specifies the format of the data contained in the edited numeric field. Listed below are the two valid entries. If neither is specified, US is assumed.

- **US** edited numeric is USA currency standard ($9,999.99)
- **IN** edited numeric is International currency standard ($9,999.99)

**N:** Numeric. This is zoned decimal format on the mainframe and display numeric on the PC. This produces the numeric format used by most applications on the PC.

**O:** Occurs. This defines the beginning of a table and must have an operation of VARTABLE or FIXTABLE. The MF size field must contain the number of times that the table occurs. For variable tables, this should be the maximum number of occurrences.

**P:** Signed Packed. This is a standard format packed number that contains a sign, either X'C' or X'D' in the low order nibble (nnnnnC). It is display numeric on the PC.

**S:** Always Signed. This is zoned decimal format on the mainframe and display numeric on the PC with a leading external sign.

**T:** Transparent. Indicates that the field will be transmitted with no data conversion taking place. This should be used for transferring packed fields between the PC and mainframe when they will be packed on the PC.

**U:** Unsigned Packed. This is a packed number that always has a X'F' in the low order nibble (nnnnnF). It is display numeric on the PC.

**W:** Packed No Sign. This is a packed number with no sign nibble. The low order digit occupies the low order nibble (nnnnnn). It is display numeric on the PC.

**Z:** Zoned Decimal. Standard mainframe zoned decimal format. On the PC it adheres to the zoned decimal format used by the MicroFocus/Microsoft/IBM COBOL compiler.

**Dates:** If the Output Data Type is "WKS", "Excel", or "xBASE", the following logical data types may be used in the last two positions of the MF Type field to indicate that the year is to be expanded to include the century when downloading (if the century is NOT included in the mainframe data record). The century will be determined by coding the parameter CUTOFF=nn in the Value field, or if not coded, the cutoff year will be acquired from the Options Record. The logical format of the data in the field. (See Chapter Five – Managing pcMAINFRAME, Options and Configuration, for further explanation of the Assume 19xx Cutoff Year parameter).

- **D1** date in YYDDD format
- **D2** date in CCYYDDD format
- **D3** date in MMDDYY format
- **D4** date in MMDDCCYY format
- **D5** date in DDMMYY format
- **D6** date in DDMMCCYY format
- **D7** date in YYMMDD format
- **D8** date in CCYYMMDD format
- **D9** date in CYYDDD format

If the Output Data Type is FIXED, ASCII, or BASIC, the following logical data types may be used in the last two positions of the MF Type field to indicate that the year is to be expanded to include the century when downloading. The century will be determined by coding the parameter CUTOFF=nn in the Value field, or if not coded, the cutoff year will be acquired from the Options Record. (See Chapter Five – Managing pcMAINFRAME, Options and Configuration, for further explanation of the Assume 19xx Cutoff Year parameter).

- **X1** date in YYDDD format
- **X2** date in MMDDYY format
- **X3** date in DDMMYY format
- **X4** date in YYMMDD format

**PC Size:** this is the length of the field in the record on the PC. Entries in this field are valid only when the **Output Data Type** is Fixed, WKS, ASCII, Excel or xBASE. Other data types are variable in length and will adjust the length to the actual data length. **If this field is left blank,** the PC Size defaults to the MF Size.

In "WKS" or “Excel” downloads, the column width is set to the PC Size. This is valid for character fields but may cause difficulties on numeric fields. On output, they are converted to a character format for use at the PC and they expand in length. A 5 byte packed field with two decimal places and a negative value may require up to 11 bytes in the output record and a two byte binary number may require up to 7 bytes in the output record. Example x'172533763D' (5 bytes) expands to -1725337.63 (11 bytes).

**PC Type:** this field is made up of two parts and is valid only if the **Output Data Type** is WKS, Excel, xBASE or Btrieve. The first part is the data type on the PC and is required. The second part describes the attributes or field format of the PC data and is valid for Excel, WKS and Btrieve. Valid Data Types are:

- **C** Character or label fields
- **D** date fields
- **F** floating point numeric fields for xBASE only
- **L** logical type fields for xBASE only
- **M** memo fields for xBASE only
- **N** numeric fields
• **Time data type (Btrieve only)**

The second part of this field describes the display attributes given to the field during a download. This does not affect the format that the data is stored in on the file.

Valid Numeric Attributes for Excel and WKS only:

- **N1** currency attribute ($999,999.00)
- **N2** percent field (99.99%) – **Note:** when setting the number of decimal places, pcMAINFRAME uses the decimal value in the profile and subtracts two if the value is greater than two.
- **N3** scientific notation (x.xxE+xx)
- **N4** comma separated (999,999.00)

Valid Date attributes for Excel and WKS only:

- **D1** date attribute for (01-JAN-92)
- **D2** date attribute for (01-JAN)
- **D3** date attribute for (JAN-92)
- **D4** date attribute for "INTERNATIONAL LONG"
- **D5** date attribute for "INTERNATIONAL SHORT"

There are additional numeric and character data types that apply only to Btrieve data sets. See Appendix A for Btrieve data type coding if you are defining a Btrieve transfer.

**Dec Plc:** this entry is invalid for non-numeric entries, type **C** and **T**. The number of decimal places may range from 0 to 18. If no entry is made, 0 is assumed.

**Operation:** this **required** field specifies the function to be performed in creating the output record. On upload, no summarization or calculation takes place; only reformatting. The valid operations are:

- **All:** this transfers the entire record with no field formatting, therefore only TEXT data can be shipped unless transparent output data format is used. No padding or truncation takes place so the input record length is preserved. The only other entry permitted with ALL is **KEY**. This may be added to specify the location of a record key for KSDS uploads. The use of **KEY** is required in cases where the record key begins in a location other than the first position of the record.

- **Average:** this places the average value of the input field within a break into the downloaded output record.

- **Count:** this places a count of the number of records within the break into the downloaded output record.

- **Endtable:** this defines the end of either a Fixtable group or a Vartable group. pcMAINFRAME uses this position to calculate the length of an occurrence.

- **First:** this uses the first input field value within a break for the downloaded output value.

See Defining Tables in this chapter for further explanation.
**Fixtable:** this defines the beginning of a group of fields that occurs a fixed number of times. The Begin Pos field on this entry's line identifies the position in the record where this group starts. The MF Type indicates "O" for occurs and the MF Size is the number of occurrences of this group of fields. There must be an Endtable operation at the end of the group. Fields defined within the group specify their Begin Pos relative to the group, so the first field in the group has a Begin Pos of 1. Fixtable groups may occur anywhere in the output format and there may be multiple Fixtable groups.

**Insert:** this places the value placed in the next field **Picture/Fixed-text** into the output record. This facility may be used for all data types during upload and download. A number of global variables are available for inserting commonly used items into an output record. These items are placed into the output record by placing the global variable name into the Picture/Fixed-text field on an insert operation. Constant global variables (i.e. LOW-VALUE) will support a field length up to 253 bytes. The global variable names are:

- `=any character` - this will generate a field filled with the character specified
- `=DATE` - this will generate the current date in MMDDYY format.
- `=DATEL` - this will generate the current date in MMDDCCYY format
- `=HIGH-VALUE(S)` - this will generate a field containing x'FF's
- `=IDATE` - this will generate the current date in DDMMYY format.
- `=IDATEL` - this will generate the current date in DDMMCCYY format
- `=JDATE` - this will generate the current Julian date.
- `=JDATEL` - this will generate the current Julian date including century designation (ccyyddd).
- `=LOW-VALUE(S)` - this will generate a field containing x'00's
- `=PC ID` - this will generate the ID of the PC issuing the request.
- `=SPACE(S)` - this will generate a field of blanks
- `=TASKNO` - this will generate the task number of the current pcMAINFRAME session
- `=TIME` - this will generate HHMMSS of the start time of the current pcMAINFRAME session
- `=ZERO(S)` - this will generate a field of zeros in a character field.

If you wish to create a numeric field valued zero, use INSERT as the operation, and place a 0 in the Picture/Fixed-text field

**Note:** if a profile contains one or more **Insert fields** preceding any selected fields and if it is used for both upload and download, the data uploaded will not be the equivalent of the data downloaded unless there is a "Skip" operation following each "Insert" operation. The insert will operate in both directions. Thus, the uploaded file will have an additional field that was not present on the downloaded file unless the "Skip" is also specified.
Key: this identifies the position of the record key for a KSDS file. It does not define a field to be uploaded or downloaded. If this identifier is not present, the record key is assumed to begin in the first position of the record. Use Key to identify keys that begin in positions other than position one of the record.

Move or Last: this is the most commonly used operator and will cause the last value of the input field before a break to be moved to the output record -- this is copying the input field to the output field. This facility may be used during upload and download.

Maximum: this places the greatest value of the input field within the break in the downloaded output record.

Minimum: this places the least value of the input field within the break into the downloaded output record.

Sequence: inserts a field containing a sequential number. This number starts at one and increments by one with each record output.

Skip: a method of not using or of bypassing selected fields of data being uploaded from the PC. The field in the position described by the SKIP entry will not be included in the record written to the mainframe. Skip is ignored for download operations.

Sum: this places the sum of the input field within the break into the downloaded output record.

Variance: this calculates the statistical function variance using the input fields within the break as the "population" and places it in the downloaded output record. "Varsamp" may be used to calculate variance using the input amounts as a "sample" rather than the "population."

Vartable: this defines the beginning of a group of fields that occur a variable number of times. A Vartable group may only be defined at the end of a record and there may only be one of them. The Begin Pos field on the Vartable entry defines the beginning position in the record of the first occurrence of this group. The MF Type indicates "O" for occurs and the MF Size is the maximum number of occurrences of this group of fields.

Picture / Fixed-text: this optional entry contains the character representation or figurative constant of data to be inserted in the output record if the operation is "INSERT." Otherwise, it is a field displayed by pcMAINFRAME to provide notes, messages and comments. This field is valid for both uploads and downloads.

Defining Tables

Three different field description lines are needed to define a TABLE to pcMAINFRAME. They are:

- **FIXTABLE/VARTABLE** line to specify the beginning of the table.
- One or more DATA lines to define the field(s) in the table
- The ENDTABLE line to signal the end of the table definition.

Only the fields in the table that are to be transferred need to be defined. Fields not defined will be not be transferred, but all occurrences of the table will be transferred.
For variable tables, the length of the input record will determine the number of occurrences that are transferred.

It is possible to have only a single occurrence or a range of occurrences extracted from a table. This is known as selective table occurrence processing. The "from" and "to" occurrence numbers can be specified as constant values or as variables.

For uploads, it is assumed that the PC file does not contain the full table, but only the fields in the selected range.

The following details the coding rules for each of the table definition lines:

**FIXTABLE/VARTABLE** for defining the beginning of the table:

- **Data/Field Name:** Any field name to describe the table.
- **Begin Pos:** The location in the mainframe record that the first field in the first table occurrence begins.
- **MF Size:** The number of occurrences in the table. If variable, the maximum number of occurrences.
- **MF Type:** The type is always "O" (Occurs)
- **PC Size:** Not valued.
- **PC Type:** Not valued.
- **Dec Plc:** For selective occurrence processing only, the selected starting occurrence number.
- **Operation:** Either FIXTABLE or VARTABLE.
- **Picture/Inserted Data:** For selective occurrence processing only, a VARIABLE NAME that will be used to prompt for the starting occurrence number (e.g. &BEGIN_OCCURRENCE)

**DATA** to define data lines:

All data fields should be entered just like any other field with the following exception:

- **Begin Pos:** The location of the field in the mainframe record relative to the beginning of the table. Thus, the first field would begin in position 1.

**ENDTABLE** to define the end of the table definition:

- **Data/Field Name:** Any field name to indicate the end of the table.
- **Begin Pos:** The location in the mainframe record that the first field of the second occurrence of the table begins.

For example, a group containing three 15 byte fields that start in position 30 of a record would have 75 as the Begin Pos of the Endtable entry. 30 (Beg Pos) + 45 (length of one occurrence) = 75 (next byte after first occurrence). pcMAINFRAME uses this position to calculate the length of an occurrence.

- **MF Size:** Not valued.
- **MF Type:** Not valued.
- **PC Size:** Not valued.
- **PC Type:** Not valued.
Dec Plc: For selective occurrence processing only, the selected ending occurrence number. If a starting number is defined and this field is left blank, then only one occurrence will be transferred.

Operation: ENDTABLE.

Picture/Inserted Data: For selective occurrence processing only, a VARIABLE NAME that will be used to prompt for the ending occurrence number (e.g. &END_OCCURRENCE).

PC and User ID Maintenance

The third CICS menu item available is PC ID Maintenance. In the TSO version this is called User ID Maintenance. Throughout this section PC ID and User ID will be used interchangeably, except as specifically noted. The term ID may be used to apply to either. ID records are used to identify authorized users of pcMAINFRAME and to establish valid class codes and library data sets for each user. The content of the screen varies slightly from the CICS to the TSO version.

When this function is initially invoked, the first ID field is blank. If you wish to display, update, or delete an existing ID, or add a new ID by modeling from an existing ID, type the existing ID and press the Enter key. The screen will then display the ID’s attributes. A sample of the CICS screen is shown in Figure 4-15 and the TSO screen is in Figure 4-16.

<table>
<thead>
<tr>
<th>(pcMAINFRAME)</th>
<th>PC ID Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC ID ............</td>
<td>DEMO</td>
</tr>
<tr>
<td>Description ....</td>
<td>Sample PCID</td>
</tr>
<tr>
<td>Password ..........</td>
<td>DEMO</td>
</tr>
<tr>
<td>Classes ............</td>
<td>*</td>
</tr>
<tr>
<td>Private library ....</td>
<td>CFXLIB</td>
</tr>
<tr>
<td>Public library ....</td>
<td>CFXLIB</td>
</tr>
<tr>
<td>Customized View .....</td>
<td>P</td>
</tr>
<tr>
<td>Customized Save ....</td>
<td>P</td>
</tr>
<tr>
<td>Authorized Prefix ...</td>
<td>AR ACCT AP BDGT</td>
</tr>
<tr>
<td>Max IDs in Pool .....</td>
<td>000</td>
</tr>
<tr>
<td>Last maint date. ....</td>
<td>01/21/97</td>
</tr>
<tr>
<td>Last used date. ....</td>
<td>02/12/97</td>
</tr>
</tbody>
</table>

UPDATE function in progress for this ID.
Enter "D" to delete or "U" to update PC ID.

Depress Clear to return.

Figure 4-15: PC ID Maintenance Screen - CICS

<table>
<thead>
<tr>
<th>(pcMAINFRAME)</th>
<th>User ID Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID........</td>
<td>A0370RF</td>
</tr>
<tr>
<td>Description.....</td>
<td>TEST CF SOFTWARE</td>
</tr>
<tr>
<td>IndForm File OK..</td>
<td>Y</td>
</tr>
<tr>
<td>Classes..........</td>
<td>*</td>
</tr>
<tr>
<td>Private library..</td>
<td>CFXLIB</td>
</tr>
<tr>
<td>Public library..</td>
<td>CFXLIB</td>
</tr>
<tr>
<td>Customized View..</td>
<td>P</td>
</tr>
<tr>
<td>Customized Save..</td>
<td>P</td>
</tr>
<tr>
<td>Authorized Prefix</td>
<td>AR ACCT AP BDGT</td>
</tr>
<tr>
<td>Last maint date..</td>
<td>01/23/97</td>
</tr>
<tr>
<td>Last used date...</td>
<td>01/27/97</td>
</tr>
</tbody>
</table>

UPDATE function in progress for this ID.
Enter "D" to delete or "U" to update ID.

PF1 - Return to directory, Clear - Menu

Figure 4-16: User ID Maintenance Screen - TSO
This screen allows you to add new IDs to the system, update existing IDs or delete IDs that are no longer needed. When the function is first initiated, a formatted screen is displayed with the cursor in the field identified as ID.

The data displayed for each PC ID is described below.

**PC ID:** CICS only. **Required.** A unique 4 character identifier for each PC accessing pcMAINFRAME. All reports and logs identify PC users by this ID. A PC ID is specified on the mainframe host and in the setup file on each PC. When the PC establishes communication with the host, its PC ID, along with the password (defined below), is validated against the PC ID defined on the mainframe.

The PC ID may be the same as a CICS operator ID. If the PC ID is made to match a CICS operator ID and the PC user puts an "*" in the PC ID field of the setup file, the password is not checked by pcMAINFRAME. Instead, pcMAINFRAME checks to make sure that the CICS operator ID for the terminal invoking pcMAINFRAME matches a valid PC ID. If it does, that PC ID is used for the pcMAINFRAME session. If the CICS operator ID does not match a PC ID, the session is terminated. The CICS operator ID comes from the OPIDENT parameter of the signon table macro. If you use the VSE interactive interface instead of the CICS signon table, the CICS operator ID comes from the first 3 characters of the USERID defined to the interactive interface. In this case, validation is left up to the CICS signon facility.

**User ID:** TSO only. **Required.** The 7 character TSO User ID identifies the TSO session to be used for pcMAINFRAME functions. This ID will be used to identify the user throughout the system. When a pcMAINFRAME session begins, it receives the ID of the TSO user logged on to the current session from the host and validates it against IDs defined to pcMAINFRAME. If the IDs match, the attributes on this screen are used during the session. If there is no matching ID defined to pcMAINFRAME, further processing is prohibited.

**Description:** this is an **optional** 35 byte field that describes the PC ID with which it is associated. It appears in the PC ID Directory and is purely for your convenience.

**Password:** CICS only. **Required.** A required, 1 to 8 character string associated with the PC ID. Each time a PC initiates a communication session with the mainframe, the user is prompted to enter this password. Failure to correctly enter it in three attempts terminates the communication. This field is mandatory but the entries need not be unique from PC ID to PC ID.

**IndForm File OK:** TSO only. This option permits the use of an individual form file by this user. If this option is set to Y, the user may define and use profiles in a private form file to be used exclusively by this user.

**Classes:** 1 to 8 alphabetic entries (A-Z) to indicate which profile classes this PC ID is authorized to access. A value of asterisk (*) will allow access to all classes.

**Private Library:** this is the FCT name of the pcMAINFRAME private library file to be used by this ID. This is an **optional** feature and if you do not want a user to have access to the private library upload feature, specify "NONE" as the private library entry. The default name is
"CFXLIB" which is the Library file shipped with pcMAINFRAME. If the library field is an asterisk (*), the PC ID may upload to any library for which it has profile access. Downloads may be done from any library for which a PC ID has profile access, regardless of the library field contents in its PC ID record. Members in private libraries may be password protected to prevent unauthorized access to the member.

**Public Library:** this is the FCT name of the pcMAINFRAME public library file to be used by this PC. This is an optional feature and if you do not want a PC to have access to the public library upload feature, specify "NONE" as the public library entry. The default name is "CFXLIB" which is the Library file shipped with pcMAINFRAME. If the library field is an asterisk (*), the PC ID may upload to any library for which it has profile access. Downloads may be done from any library for which a PC ID has profile access, regardless of the library field contents in its PC ID record. Members in the public library are for general use and may not be password protected.

**Customized View:** this is one of the authorization limits for PC Users that use Profile Customization. If not specified, the user will be able to view any profile valid for his class. Valid designations are: P - allow access to those profiles that match the assigned prefix (see Authorized Prefix, below), and N - allow NO access for viewing profile customization.

**Customized Save:** this authorization limit for Profile Customization by PC Users affects their ability to save their User profiles. If not specified, the user will be able to SAVE to any name valid for his class. The two other designations are: P - allows saving to only those names starting with their authorized prefix, and N None - does NOT allow saving of a customized profile.

**Authorized Prefix:** One of the Profile Customization authorization limits, this allows the systems administrator to allow users access to only those profiles beginning with the alphabetic prefix designated. Each prefix can be up to 4 bytes in length, and 4 prefixes are allowed. This field works in conjunction with the Customized View and Customized Save parameters. If no authorized prefixes are entered, the limitations are dependant strictly upon the Classes allowed.

**Max IDs in Pool:** CICS only. An optional feature which allows up to 999 concurrent users of the same PC ID. This feature enables the systems administrator to assign a single PC ID for a group of users, perhaps in the same department. Transaction logs will show the PC ID appended with a sequence number, as in ACCT1, ACCT2, ACCT3, etc., up to 999. The Library facility will show only the 4 position PC ID (ACCT).

All fields may be entered or updated by typing the new information into the appropriate fields. The cursor may be moved to any field. Press Enter when changes are complete; pcMAINFRAME will then edit the input. If any errors were present, the appropriate message will be displayed. Correct any errors until a message prompts you to: Enter "D" to delete or "U" to update PC ID, or Enter "A" to add PC ID. If you wish to perform one of those functions, enter the correct letter in the update selection field at the bottom of the screen and press Enter. If you do not wish to perform one of those functions, press Close to return to the main menu.
Profile Directory

Entering this function displays a blank Profile Directory screen with the cursor positioned in the name field. The name field is generic, meaning that a full or partial profile name may be entered. The display that follows will begin with the first profile name that is equal to or greater than the name, or partial name, that was entered.

Type into the name field the full or partial name of the profile at which you wish to begin the directory display and press \Enter. If the name field is left blank, the directory will begin with the first profile in the file. The display of the directory is shown in Figure 4-17 below.

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>Profile Class</th>
<th>Dataset Name</th>
<th>Dataset Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFXESDS</td>
<td>*</td>
<td>CFXESDS</td>
<td>KSDS</td>
<td>Fixed</td>
</tr>
<tr>
<td>DEMO1</td>
<td>*</td>
<td>CFXESDS</td>
<td>KSDS</td>
<td>Basic</td>
</tr>
<tr>
<td>LIB</td>
<td>*</td>
<td>CFXLIB</td>
<td>Lib</td>
<td>UP OR DOWN LOAD FILE</td>
</tr>
<tr>
<td>RIB</td>
<td>*</td>
<td>CFXLIB</td>
<td>Lib</td>
<td>UP OR DOWN LOAD LIB - WORD PROCESS</td>
</tr>
<tr>
<td>POWER01</td>
<td>*</td>
<td>CFPOWEX</td>
<td>Exit</td>
<td>Spool</td>
</tr>
<tr>
<td>POWER02</td>
<td>*</td>
<td>CFPOWEX</td>
<td>Exit</td>
<td>Spool</td>
</tr>
<tr>
<td>POWER03</td>
<td>*</td>
<td>CFPOWEX</td>
<td>Exit</td>
<td>Fixed</td>
</tr>
<tr>
<td>POWER04</td>
<td>*</td>
<td>CFPOWEX</td>
<td>Exit</td>
<td>Fixed</td>
</tr>
<tr>
<td>POWER05</td>
<td>*</td>
<td>CFPOWEX</td>
<td>Exit</td>
<td>Fixed</td>
</tr>
<tr>
<td>POWER06</td>
<td>*</td>
<td>CFPOWEX</td>
<td>Exit</td>
<td>Trans</td>
</tr>
<tr>
<td>POWER07</td>
<td>*</td>
<td>CFPOWEX</td>
<td>Exit</td>
<td>Trans</td>
</tr>
<tr>
<td>POWER08</td>
<td>*</td>
<td>CFPOWEX</td>
<td>Exit</td>
<td>Fixed</td>
</tr>
<tr>
<td>POWER09</td>
<td>*</td>
<td>CFPOWEX</td>
<td>Exit</td>
<td>Fixed</td>
</tr>
<tr>
<td>SALES1</td>
<td>A</td>
<td>CFXKSDS</td>
<td>KSDS</td>
<td>DIF</td>
</tr>
</tbody>
</table>

Figure 4-17: Profile Directory

The fields that are shown consist of:

Profile Name: this is the one to eight character name by which the profile is known.

Profile Class: this is the class associated with this profile and matched against PC ID classes.

Dataset Name:

CICS: the 1 to 7 character FCT name of the file that this profile accesses.

TSO: the 1 to 8 character ddname or .ddname of the dataset that this profile accesses. If the .ddname form is displayed this indicates that a ddrec is used and the dataset is dynamically allocated.

Dataset Type: this indicates the structure of the file accessed by this profile.

Data Type: either ASCII, Basic, Excel, DIF, Fixed, Include, Trans, Spool, WKS, or xBASE to indicate the data type of the file on the PC used by this profile.

Description: a text description created by the administrator when the profile was added or updated.

The directory display can be advanced to additional pages, if they are present, by pressing /G0B/G33/G29/G15/G0C or /G0B/G26/G4F/G48/G44/G55/G0C will return the display to the menu screen. At any time
you may type a new, partial, or full profile name and press \Enter\ to reposition the directory display.

**Update from the profile Directory**

If you wish to update a profile that is displayed on the directory, position the cursor, by using the \Enter\ or arrow keys, to the line containing the profile name and press the \Enter\ key. This will transfer the screen directly to function 2, Profile Maintenance. From this screen you can make any desired changes. Pressing \F1\ from the Profile Maintenance screen will return you to the Profile Directory screen.

**PC ID and User ID Directory**

This function displays a directory of IDs that are defined to the system. When this function is invoked, a formatted directory screen with no ID information on it is displayed. The cursor is positioned in the ID field. This field is generic, meaning that you may enter a full or partial ID and the display that follows will begin with the PC ID that you entered or the one that is next alphabetically. The screen that is displayed in CICS is shown in Figure 4-18, and the TSO screen is shown in Figure 4-19.

![Figure 4-18: PC ID Directory](pcMAINFRAME)

**PC-ID Directory**

<table>
<thead>
<tr>
<th>ID:</th>
<th>ACT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC-ID</td>
<td>Password</td>
</tr>
<tr>
<td>ACT2</td>
<td>COUNTER</td>
</tr>
<tr>
<td>ADV1</td>
<td>PROMO</td>
</tr>
<tr>
<td>DMO</td>
<td>DEMO</td>
</tr>
<tr>
<td>MKT1</td>
<td>CUBS</td>
</tr>
<tr>
<td>MKT2</td>
<td>SOX</td>
</tr>
<tr>
<td>FC01</td>
<td>BULLS</td>
</tr>
<tr>
<td>FC02</td>
<td>SEARS</td>
</tr>
</tbody>
</table>

*End*

PF1 - Menu, PF2 - Next Page, ENTER - Update Entry at cursor.

![Figure 4-19: User ID Directory](pcMAINFRAME)

**User ID Directory**

<table>
<thead>
<tr>
<th>ID:</th>
<th>A0370CF</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID</td>
<td>Classes</td>
</tr>
<tr>
<td>User ID</td>
<td>Library</td>
</tr>
<tr>
<td>ACCT010</td>
<td>A F R G</td>
</tr>
<tr>
<td>ACCT017</td>
<td>A F R G</td>
</tr>
<tr>
<td>ADVT006</td>
<td>M G</td>
</tr>
<tr>
<td>HUMR013</td>
<td>H F G</td>
</tr>
<tr>
<td>LEGL004</td>
<td>L G</td>
</tr>
<tr>
<td>LEGL006</td>
<td>L G</td>
</tr>
<tr>
<td>MGTK003</td>
<td>M G</td>
</tr>
<tr>
<td>MKTG004</td>
<td>M G</td>
</tr>
<tr>
<td>SALE003</td>
<td>M G</td>
</tr>
<tr>
<td>SALE004</td>
<td>M G</td>
</tr>
<tr>
<td>SYST001</td>
<td>*</td>
</tr>
<tr>
<td>SYST010</td>
<td>T U V W X Y Z</td>
</tr>
<tr>
<td>SYST013</td>
<td>T U V W X Y</td>
</tr>
<tr>
<td>SYST014</td>
<td>T U V W X Y</td>
</tr>
</tbody>
</table>

*End*

PF1 - Menu, PF2 - Next Page, PF3 - Prior Page, ENTER = Update Entry at cursor.

The fields that are shown consist of:

**PC ID:** this is a unique 4 character identifier associated with each PC in the system. This is the name by which each PC is known. This matches the User ID field in the setup file on the PC.
User ID: this is a unique 7 character identifier associated with each authorized user in the system. This is the name by which each user is known to TSO. This matches the User ID field in the setup file on the PC.

Password: a 1 to 8 character string associated with the PC ID.

Class: a 1 to 8 alphabetic entries (A-Z) to indicate which profile classes this PC ID is authorized to access. A value of (*) asterisk will allow access to all classes.

Private Library: this is the FCT or dd name of the pcMAINFRAME private library file to be used by this PC. This is an optional feature and if you do not want a PC to have access to the Library upload feature, specify “NONE” as the private library entry.

Public Library: this is the FCT or dd name of the pcMAINFRAME public library file to be used by this PC. Like the Private Library Feature, this is optional and can be protected from PC access by specifying "NONE" as the public library entry.

Description: this is an optional 35 byte field that describes the PC ID with which it is associated. It appears in the PC ID Directory and is purely for your convenience.

The directory display may be advanced to the next page, if one is present, by pressing the key. You may return to the menu by pressing either the or key. Pressing will display the previous page if one exists. The directory display may be repositioned to a different starting point at any time by typing a new, full or partial, PC ID and pressing the key.

Update from the PC-ID directory

If you wish to update a PC ID that is displayed in the directory, position the cursor, using the , , or arrow keys, to the line containing the PC ID and press the key. This will move you directly to function 3, PC ID Maintenance. From this screen you can make any desired changes. Pressing from the PC ID Maintenance screen will return you to the PC ID Directory screen.

Library Directory — CICS Only

This function displays a directory of library members that are present on either the default library file CFXLIB or other library files as requested. When this function is invoked, a formatted directory screen with no library information on it is displayed. The cursor is positioned in the File-Name field. This field is generic, meaning that you may enter a full or partial filename and the display that follows will begin with your entry or the first valid file name following your partial filename entry. The library name field can also be entered. This field defaults to CFXLIB, but can be any valid FCT that refers to a pcMAINFRAME library file. See Chapter 7 for more information about pcMAINFRAME libraries.
User Profile Maintenance — CICS Only

This function allows users to define profiles, based upon previously defined master profiles, for download. This gives the user great flexibility in data selection and formatting, but also gives the system administrator the ability to limit user access to a subset of a file. A User Profile is created like a standard profile, except that in creating a user profile a Master Profile name is used instead of a FCT name. User profiles act in conjunction with their master profiles during download processing. Records are initially selected, summarized and formatted by the master profile, but are then passed to the user profile for further selection, summarization and formatting.

A Systems Administrator may wish to make some data (name, department, hire date, etc.) from a payroll master file generally available for download, without providing access to the more sensitive payroll data. In this case, a profile, named EMPDATA, could be set up defining all public data fields for extract. The EMPDATA profile would then be specified as the master profile in one or more user profile definitions. Each user profile would operate on records containing only those fields defined in the master profile’s output format, but would have full select, summarize and format capability against those records.

During file transfer, the processing specifications entered in both the master and user profiles are honored independently. This includes read limits, skip counts, summarization and record selection. Master profile specifications are applied first, then as records are extracted and passed on, the user profile specifications are applied.

When menu option seven or eight is entered and the enter key is pressed, two additional fields, PCID and PCID password will appear below the selection line.

Before the User Profile Maintenance can be accessed, a PC ID and password must be entered. The class code for the PC ID is used to determine which of the master and user profiles can be viewed or modified. Figure 4-23 shows the menu screen with the PC ID and PASSWORD prompt.
The screens used to add, update and delete a User Profile look and operate like standard profiles with a few exceptions:

1. Instead of the "File or Exit" name field, a "Master Profile" name must be specified. This is the name of the profile to be used as the high level record selector/extractor for this user profile. A master profile can be any standard profile with a file type of ESDS, KSDS, RRDS or Exit that user is authorized to access (see Figure 4-24).

2. The "Access Ranges and Limits" screen can be valued when the master profile file type is KSDS. The rules for defining browse ranges are the same as for a normal profile. When this screen is valued, the key ranges defined are merged with any key ranges that are defined in the master profile before the VSAM file is read. Only those records whose keys fall within the access range limits of both the master and user profiles will be returned.
For example, given the following situation:

- Master Profile Access Range = from 01 to 05
- User Profile Access Range = from 03 to 07

The access range used would be the intersection of the 2 sets, or from 03 to 05.

---

If the master profile is changed, or if the reference to a master profile name is changed, any field changes in the "master extract set" MUST be reflected in the user profile.

---

3. The Output Format screen is the same as the Output Format screen for a standard profile. When a new user profile is added, all of the fields defined in the output format section of the master profile will be forwarded to the user profile. This screen can then be modified to fit the requirements of the specific download.

It is important to note that the data the user profile operates on has been extracted by the master profile. The format of this "master extract set" is developed internally at run time based on the fields defined in the master profile.

When data is formatted for user profile processing, the master profile data type (BASIC, DIF, etc.) is disregarded and the "master extract set" is formatted as fixed. The data is then formatted according to the specification in the User Profile. The rules regarding the formatting of the "master extract set" are shown below.

- Fields in the master extract set are built in the sequence that they are defined in the master profile’s output format screen.
- Fields retain their mainframe data type and length. A 5-byte packed field in the master profile will be a 5-byte packed field in the "master extract set".
- Computed fields (fields generated with "Average", "Count", "Sum" or "Variance") will always be formatted as 10-byte packed fields.
- The beginning position and length reflect the output of the master profile. This may differ from the original data set. The values will be inserted automatically so long as the field names in the User Profile are the same as the Master Profile.

Refer to the section on Profile Maintenance for a detailed explanation of each field.
**User Profile Directory — CICS Only**

Entering this function displays a blank *User Profile Directory* screen with the cursor positioned in the name field. The name field is generic, meaning that a full or partial profile name may be entered. The display that follows will begin with the full profile name you entered or the profile name closest to the partial name you entered. A user cannot view a profile unless the profile has the same class code as one defined for the user’s PC ID.

From the name field, type the full or partial name of the User Profile you wish the directory to begin displaying and press **/G0B/G28/G51/G57/G48/G55/G0C**. See the *Profile Directory* section above for an explanation of the fields on this screen.

![Figure 4-24: User Profile Directory](image)

**DDREC Maintenance - TSO Only**

DDREC entries carry the required information about datasets that will be dynamically allocated or created by pcMAINFRAME. Entries are identified by a 7 character DDREC ID.

When this function is first selected, a blank screen is presented. Enter into the DDREC ID field either the name of an existing id that you wish to update or the name of a new id that you wish to add, and press **/G0B/G28/G51/G57/G48/G55/G0C**. Make any entries or changes and press **/G0B/G28/G51/G57/G48/G55/G0C** to edit the input. If the input is correct, enter **A** to add, **U** to update, or **D** to delete in the last field on the screen and press **/G0B/G28/G51/G57/G48/G55/G0C** to complete the operation.

![Figure 4-25: DDREC Maintenance](image)
The fields that are shown consist of:

**DDREC ID:** required this is the 1 to 7 character name by which this dataset will be referred to in profiles.

**DDName:** optional this is the ddname that will be used for the allocation. If this is not specified, a ddname of CFXDATA will be used.

**Full DSNAME:** required this is the full name of the data set being accessed. Note: Use apostrophes (’’) around the DSNAME. TSO appends your user id to the beginning of the DSNAME if it is not enclosed in apostrophes. This will cause the file not to be found when an allocation is attempted.

**File Password:** if the data set being accessed requires a password, enter it here.

The next 3 fields are only during upload processing of sequential files.

**File exists Action:** this is the action that pcMAINFRAME is to take if the file specified already exists when it is dynamically allocated for an upload. Choose one of the options:

- **FAIL:** Do not run the transfer.
- **REPLACE:** Replace the existing data with the uploaded data using the existing file DCB and space parameters.
- **RECREATE:** Delete the existing file and create a new file using the DCB and space parameters in the DDREC.
- **MOD:** Add the uploaded data to the end of the existing file. This is not valid with a PDS.

**File does not exist action:** this is the action that pcMAINFRAME is to take if the file specified does not exist when it is dynamically allocated for an upload. Choose one of the options:

- **FAIL:** Do not run the transfer.
- **CREATE:** Create a new file using the space and DCB parameters contained in the DDREC.

**Transfer fail action:** this is the action that pcMAINFRAME is to take if an error occurs during an upload. Choose one of the options DELETE, UNCATLG, or CATLG.

**Allocate exclusive:** specify Y or N to indicate whether the file is to be allocated with exclusive control during the transfer.

The following options, DeviceParms, SpaceParms, and DCBParms are used only for files that are created during uploads. If the file to which you are uploading already exists, these options are not required as the values already associated with the dataset will be used unless you specified RECREATE. These are the same parameters that you would supply for a DD statement in your JCL.

**Device parms**

- **Unit:** this is for the type of device on which the file is to be created.
- **Serial number:** this is the serial number of the tape or disk.

**Space parms**

- **Type:** for disk files specify the unit in which space will be allocated. Select BLK, TRK, or CYL.
**Allocation:** specify the number for the primary and optionally the secondary allocations. This is the number of the type units you specified in the previous entry.

**DCB parms**

**Recordformat:** specify the format of the records that will be uploaded. Select from F, V, FB, or VB.

**Lrecl:** for F or FB format records specify the length of each record.

**Blocksize:** for blocked format records specify the length or maximum length of each block.

**Description:** this is the one to four character name of the PC that created the library member.

### DDREC Directory — TSO Only

Entering this function displays a blank **DDREC Directory** screen with the cursor positioned in the id field. The id field is generic, meaning that a full or partial ddrec id may be entered. The display that follows will begin with the full ddrc id you entered or the ddrec id closest to the partial name you entered. A user cannot view a profile unless the profile has the same class code as one defined for the user’s PC ID.

From the id field, type the full or partial id of the DDREC ID you wish the directory to begin displaying and press \( \text{Enter} \). You may edit one of the ddrec entries by pressing the \( \text{/G0B/G37/G44/G45/G0C} \) key until the cursor is on the line of the entry you wish to edit and then pressing the \( \text{/G0B/G28/G51/G57/G48/G55/G0C} \) key. This will bring up the DDREC Maintenance screen as described below.

![DDREC Directory](image-url)

**EXIT pcMAINFRAME**

This function will exit the pcMAINFRAME maintenance functions and return to CICS. Pressing the \( \text{Enter} \) key will also return you to CICS.
Chapter 5 - Managing pcMAINFRAME

General Administration

Administering a pcMAINFRAME system requires periodic attention to a few different areas. Access to the system must be established and maintained; data must be qualified as accessible; PC User’s questions must be addressed; pcMAINFRAME files need backup and recovery. These are neither difficult nor lengthy tasks, but they do require some planning.

Security

Security is a primary concern in most business environments. pcMAINFRAME will introduce better avenues of security to your data access procedures. Decide how you wish to control this new access as part of your implementation. pcMAINFRAME has security classes and extensive password protection to safeguard access to itself and access to pcMAINFRAME profiles.

pcMAINFRAME access is controlled through a file of PC IDs that you maintain. Each PC ID has a password associated with it that must be manually entered by the user when a transmission is started. Users may share a PC ID that is used by a group of users, or each user may have a unique PC ID. Each PC ID is authorized to access one to eight classes of profiles. Only profiles which have a class matching a PC ID class may be used by that PC ID.

Access to individual profiles within pcMAINFRAME is also controlled by password. If a profile needs protection from general access, a unique password should be associated with it. Passwords and class structures for profiles requiring restricted access should be established according to the sensitivity of data accessed by the profile.

Classes and passwords operate together. A PC may access only a profile that is in its same class. Furthermore, if that profile is password protected, the PC user must enter the password. When an asterisk * defines PC ID class, the PC ID will have access to all classes of profiles. Likewise, if an asterisk is specified for a profile class, that profile will be accessible by all PC ID’s.
There are two additional security options with pcMAINFRAME: the security exit and the use of CICS operator security. The security exit, a middle man, is a user written program to which pcMAINFRAME passes control when a PC establishes communication with pcMAINFRAME and every time a new request is made by that PC. The security exit may determine the validity of the PC access and return to pcMAINFRAME with a code that indicates whether the function should be allowed or not.

CICS operator security may be used in place of the password security supplied with pcMAINFRAME. If the PC ID supplied by the PC from the PC setup file is an asterisk (*), pcMAINFRAME checks to see if the operator has signed on to the terminal over which pcMAINFRAME is being run. If so, pcMAINFRAME treats the CICS operator ID as the PC ID which is verified in the table of valid PC IDs. If it is valid, processing continues with that PC ID’s privileges.

Libraries

Each PC ID is optionally associated with a Public and Private library file. This is a general storage facility for the users. Different users may share a library and keep their data separate. Data in a private library is commonly accessible for download unless the user chooses to password protect his data at the time it is saved to the library. In that case, another user must know the password in order to retrieve the data. Data in a public library is not assigned a password and is therefore always commonly accessible. Multiple libraries may be established to segregate users and insert an additional layer of password protection. Decide which users should use a library for storage and assign them, according to their data needs, to public or private libraries. Any libraries other than the standard library must be created by the systems programmer and added to CICS.

PC IDs

After you have established how many and which PC IDs will be included in the system, appropriate security classes, the type of security (CICS signon or pcMAINFRAME) to be used, which PC IDs will have library storage access, and which PC IDs are to be in separate libraries. Add these PC IDs to the system using function 3 of transaction PCMM as described in Chapter 3 - Getting Started. Each PC user will need to know his PC ID, its password, its accessible profiles, and the necessary changes to the model start-up procedure for your installation. The batch program CFX750 described below can print a report of the PC IDs on your system that may be of assistance in managing these IDs. Figure 5-5 and 5-6, located in the Sample Report section, are examples of a PC ID Summary and PC ID Detail report respectively.

Backup and Recovery

The information kept in the pcMAINFRAME files is valuable and should be protected through a regular schedule of backups and tested restore procedures. There is no internal backup built into pcMAINFRAME. The internal files it uses are standard VSAM files and may be backed up and restored just like other VSAM files. You may use IDCAMS: BACKUP/RESTORE (VSE only), REPRO, or EXPORT/IMPORT; or a non-IDCAMS utility such as FAVER, Westinghouse Disk Utility, FDR, or other program that backs up VSAM files.
The files of concern are CFXFORM and CFXLIB as well as any other libraries that you may have established. These files may be journalled by CICS so that forward and backward recovery may be performed. Back up these two files at least once per day as you would other files that are updated by CICS. You may wish to have your users upload files, programs, or data from the PC to a pcMAINFRAME library for backup or archival purposes in which case a more frequent and longer retention backup scheme may be called for.

Logging and Statistics

A log of all activity in the system is maintained in two locations: in a disk log that you may review on-line from a CICS terminal, and in hard copy. During a transmission session, all informational messages between the PC and the mainframe are logged to a disk file. You may review this log with function 1 of transaction PCMM any time prior to the PC’s next transmission when it is deleted. If the history logging option is enabled, then log records will be written with a unique data/timestamp and records and can be viewed by Placing a ‘Y’ in the History? Field of the transmission log query screen.

Note: History records are aged and deleted when the retain interval in the options records has been met.

A hardcopy of the log is produced on the CICS destination CSMT or a different destination of your choosing which may be either the system printer, a disk file, or a terminal depending on your installation. Use these logs to monitor activity within pcMAINFRAME and to aid in problem determination.

If a user calls with a problem, you can use transaction PCMM function 1, Review Transmission Logs, to look at all messages exchanged between the PC and the mainframe during the last transmission. This can be helpful in determining what actually happened. The hardcopy log is valuable as an audit trail as it lists in chronological order all activity with the individual messages identified with the appropriate PC ID. Summarized statistics for each PC are also printed at the end of each session.

Controlling Profiles

Profile Definitions

Establishing and changing profiles is a key function in the control of pcMAINFRAME. The parameters you set up determine how easily users can understand the system and perform transfers, how securely corporate data is kept, and how efficiently pcMAINFRAME processes the requests.

As you work with users to determine requirements for downloading information, keep in mind the range of alternatives available in meeting their needs. The simplest and most straightforward example would be one in which data existed on the mainframe in a small file that contained exactly the data that was required by the user. Setting up a profile would then be a matter of identifying the file that contained the data and downloading it in its entirety. Matters get significantly more involved when other factors enter the picture such as: many kinds of user requests against different files, requests for different portions of a file at different times, very large files, etc. Procedures that address these problems, as well as alternative solutions, may be considered as follows:
• **Establish browse ranges** to reduce the amount of data that is processed during a request from the PC. For example, browse ranges would insure that when the manager of Plant One in Chicago performs a budget download, he reads only the data relevant to his plant instead of searching the entire file for Plant One records. The larger the file, the greater the potential savings this feature has. The browse range facility can also be used for security purposes to limit users to specified portions of a file.

• **Use record selection** to include or exclude records meeting specified criteria. Using our example above, a profile can be created so that the manager in Chicago receives budget records only and does not receive other types and, further, so that budget records for the first quarter only are selected.

• A **variable parameter facility** in profiles allows pcMAINFRAME to prompt the PC user for selection criteria during the actual selection and browsing processes instead of using a constant value defined in the profile. This would allow our Chicago manager to specify any quarter or month he wished to download during a transfer. Whenever possible, use the variable parameter facility to make profiles more general. This will allow multiple users to share the same profile for slightly different uses or the same user to change the things that vary from one use to the next. Specify passwords and classes for profiles that should not have general access. Use function 2 of transaction PCMM to create, modify, or delete profiles as explained in Chapter 3 - Getting Started.

• Use **CFX750 - Profile Report** to keep track of profiles and authorized users. This facility allows you to print either Summary or Detail information on one or more profiles during each execution, and to print PC ID information.

**Master and User Profiles**

You may give some of your users the ability to define their own profiles while still maintaining control over what data they may access. This is done through the Master/User profile facilities. Users authorized to create user profiles may define profiles that access data provided by other (Master) profiles rather than a data set. You can define profiles, and therefore subsets of master data, that users use as a source from which their profiles extract data. This facility gives you control over data access while it provides users with the flexibility to define their own selection and formatting options.

**Profile Report - CFX750**

To assist in administration, pcMAINFRAME provides CFX750, the Profile Report, a batch program for printing profiles and PC IDs. A summary report produces a one line description of each profile, while a detail report shows all the fields associated with a profile. Samples of both types of reports, for both profiles and PC IDs, follow the control card syntax.
**Parameter Card Syntax**

CFX750 uses two input parameter cards to control its processing. These cards must begin in Column 1 and be in the following order: **REPORT** first, followed by **LINE**, if you wish to use the second card.

```
REPORT= [ BOTH | SUMMARY | DETAIL [ (profile1,profile2) ] | PCID | ALL ]
[LINEnnn]
```

The first control card controls the type of report to produce (Summary or Detail) and which profile to print. There are six possible values that may be used for the first control card. They are:

- **REPORT**=
  - **BOTH**: Both a Summary and Detail report will be produced for all profiles on file. This is the default if no parameter card is present.
  - **SUMMARY**: a Summary report will be produced for all profiles on file.
  - **DETAIL**: a Detail report will be produced for all profiles on file.
  - **DETAIL(profile1,profile2)**: a Detail report is produced for each profile listed.
  - **PCID**: This will produce a summary and detail PC ID report.
  - **ALL**: Produces all of the reports: Summary and Detail PC ID reports and Summary and Detail Profile reports.

- **LINE**=

  The second control card is optional and controls the number of lines per page to print. If this card is left out, the report will be produced using 56 lines per page. The value for this parameter is:

  - **nnn**: The number of lines per page

**Sample JCL**

The following figures show sample JCL to run the profile report.

![Figure 5-1: Sample VSE JCL](image)

```
// JOB FORMPRNT
// DLBL CFXFORM, 'PCMAIN.FRAME.PROFILE', VSAM,CAT=SYSACT
// ASSIGN SYS090, SYS10
// EXEC CFX750
REPORT=DETAIL(SALES1,CFXESDS)
/*
```

*Figure 5-1: Sample VSE JCL*

![Figure 5-2: Sample MVS JCL](image)

```
//FORMPRNT JOB ..... 
// * - - - - PROFILE REPORT
//STEPLIB EXEC PGM=CFX750
//STEPLIB DD DSN=PCM.LOADLIB.LOAD,DISP=SHR
//CFXFORM DD DSN=PCMAIN.FRAME.PROFILE,DISP=SHR
//REPORT DD SYSOUT=* 
//REPORT=DETAIL(SALES1,CFXESDS)
/*
```

*Figure 5-2: Sample MVS JCL*
Sample Reports

The following are sample Profile Reports.

### Sample Profile Reports

#### 08/10/99 pcMAINFRAME SUMMARY PCID REPORT

<table>
<thead>
<tr>
<th>PCID</th>
<th>PASSWORD</th>
<th>CLASSES</th>
<th>PRIVATE LIB</th>
<th>PUBLIC LIB</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABLX</td>
<td>CLIFF</td>
<td>ABCDEF</td>
<td>*</td>
<td>NONE</td>
<td>XXXX</td>
</tr>
<tr>
<td>ADV1</td>
<td>PROMO</td>
<td>XGZ</td>
<td>CFXLIB</td>
<td>NONE</td>
<td>ADVERTISING DEPT ID</td>
</tr>
<tr>
<td>DEMO</td>
<td>DEMO</td>
<td>*</td>
<td>CFXLIB</td>
<td>NONE</td>
<td>THIS IS FOR DEMONSTRATIONS</td>
</tr>
<tr>
<td>FIN1</td>
<td>SAIL</td>
<td>GFM</td>
<td>CFXLIB</td>
<td>NONE</td>
<td>BOB C. FINANCE MINISTER</td>
</tr>
<tr>
<td>MKT1</td>
<td>CUBS</td>
<td>ABMT</td>
<td>CFXLIB</td>
<td>NONE</td>
<td>JOHN D. MARKETING DEPT.</td>
</tr>
<tr>
<td>PC01</td>
<td>BULLS</td>
<td>*</td>
<td>CFXLIB</td>
<td>NONE</td>
<td>GENERAL USER NUMBER 1</td>
</tr>
</tbody>
</table>

**Figure 5-3: Summary PCID Report**

#### 08/10/99 pcMAINFRAME DETAIL PCID REPORT

<table>
<thead>
<tr>
<th>PCID ......................... DEMO</th>
<th>PCID DESCRIPTION: THIS IS FOR DEMONSTRATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASSWORD ..................... DEMO</td>
<td>CLASSES: *</td>
</tr>
<tr>
<td>PRIVATE LIB .................. CFXLIB</td>
<td>PUBLIC LIB: NONE</td>
</tr>
<tr>
<td>LAST MAINT DATE ................07/14/97</td>
<td>LAST USED DATE: 03/23/98</td>
</tr>
<tr>
<td>ACCESSIBLE PROFILES:</td>
<td></td>
</tr>
<tr>
<td>CFXESDS DEMO1 LIB LIBTRAN LIBW POWERU1 POWER01 POWER02 POWER03 POWER04 POWER05 POWER06 POWER07 POWER08 POWER09 SALES1</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5-4: Detail PCID Report**

#### 08/16/99 pcMAINFRAME PROFILE SUMMARY REPORT

<table>
<thead>
<tr>
<th>PROFILE NAME</th>
<th>CLASS</th>
<th>DESCRIPTION</th>
<th>TYPE</th>
<th>DIRECTION</th>
<th>CHANGED</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFXESDS</td>
<td>*</td>
<td></td>
<td>FIXED</td>
<td>DOWNLOAD</td>
<td>00/00/00</td>
</tr>
<tr>
<td>DEMO1</td>
<td>*</td>
<td>DOWNLOAD 133 DATA FOR TEMPLATE ABC</td>
<td>KSDDS</td>
<td>MIX</td>
<td>DOWNLOAD</td>
</tr>
<tr>
<td>LIB</td>
<td>*</td>
<td>UP OR DOWN LOAD LIBRARY FILE</td>
<td>CFXLIB</td>
<td>MIX</td>
<td>DOWNLOAD</td>
</tr>
<tr>
<td>LIBTRAN</td>
<td>*</td>
<td>TRANSPARENT MODE LIBRARY TRANSFER</td>
<td>CFXLIB</td>
<td>MIX</td>
<td>UPLOAD</td>
</tr>
<tr>
<td>LIBW</td>
<td>*</td>
<td>UP OR DOWN LOAD LIB - WORD PROCESS</td>
<td>CFXLIB</td>
<td>MIX</td>
<td>UPLOAD</td>
</tr>
<tr>
<td>POWERU1</td>
<td>*</td>
<td>DOWNLOAD A REPORT FROM VSE/POWER</td>
<td>CFXPOwx</td>
<td>EXIT</td>
<td>SPool</td>
</tr>
<tr>
<td>POWER01</td>
<td>*</td>
<td>DOWNLOAD A REPORT FROM VSE/POWER</td>
<td>CFXPOwx</td>
<td>EXIT</td>
<td>SPool</td>
</tr>
<tr>
<td>POWER02</td>
<td>*</td>
<td>UPLOAD A REPORT TO VSE/POWER</td>
<td>CFXPOwx</td>
<td>EXIT</td>
<td>SPool</td>
</tr>
<tr>
<td>POWER03</td>
<td>*</td>
<td>DOWNLOAD PUNCH FROM VSE/POWER</td>
<td>CFXPOwx</td>
<td>EXIT</td>
<td>FIXED</td>
</tr>
<tr>
<td>POWER04</td>
<td>*</td>
<td>UPLOAD PUNCH TO VSE/POWER</td>
<td>CFXPOwx</td>
<td>EXIT</td>
<td>FIXED</td>
</tr>
<tr>
<td>POWER05</td>
<td>*</td>
<td>UPLOAD JOBSTREAM TO VSE/POWER</td>
<td>CFXPOwx</td>
<td>EXIT</td>
<td>FIXED</td>
</tr>
<tr>
<td>POWER06</td>
<td>*</td>
<td>UPLOAD BINARY FILE TO POWER PUN</td>
<td>CFXPOwx</td>
<td>EXIT</td>
<td>TRANS</td>
</tr>
<tr>
<td>POWER07</td>
<td>*</td>
<td>DOWNLOAD</td>
<td>CFXPOwx</td>
<td>EXIT</td>
<td>TRANS</td>
</tr>
<tr>
<td>POWER08</td>
<td>*</td>
<td>DOWNLOAD A QUEU LIST FROM VSE/POWER</td>
<td>CFXPOwx</td>
<td>EXIT</td>
<td>FIXED</td>
</tr>
<tr>
<td>POWER09</td>
<td>*</td>
<td>ISSUE DISPLAY CMD/DOWNLOAD RESULTS</td>
<td>CFXPOwx</td>
<td>EXIT</td>
<td>FIXED</td>
</tr>
<tr>
<td>SALES1</td>
<td>A</td>
<td>DEMO - SALES BY BRANCH &quot;TEST&quot;</td>
<td>CFXESDS</td>
<td>KSDDS</td>
<td>DIF</td>
</tr>
</tbody>
</table>

**Figure 5-5: Profile Summary Report**
PROFILE NAME........... SALES1
PROFILE DESCRIPTION..... DEMO - SALES BY BRANCH *TEST*
PROFILE CLASS......... A
PASSWORD (OPTIONAL)..... NONE
FILE OR EXIT NAME....... CFKSDS
FILE TYPE............... KSDS
OUTPUT DATA FORMAT..... DIP
TRANSLATION MODES...... MIXED CASE
WRITE CTRL-Z AT EOF..... YES
READ LIMIT............. 12345
SKIP BEFORE PROCESSING.. 0
MULTIPLE RECORD HEADER.. N
FIXED LENGTH............ 0

PCID(S):

ACCESS RANGES & LIMITS.. NONE

RECORD SELECTION:

ACCESS RANGES & LIMITS.. NONE

MULTIPLE FILE DEFINITION.. NONE

USER EXIT PARAMETER..... NONE

OUTPUT FORMAT:

Figure 5-6: Detail Profile Report
Options Configuration

There are many options that may be selected to tune and customize your installation. Some of these options include transmission block size, transmission log options, security for the administrative functions, and the language in which all mainframe messages and activity will be displayed.

These options can be configured with the Options Customization screen. This facility does not appear on the main menu, but it may be entered from the PCMM menu by pressing [G0B/G33/G29/G1A/G0C]. Figure 5-7 is an example of the Options Customization screen for CICS and figure 5-8 is an example of the screen for TSO.

The following fields may be changed:

**Class Codes:** CICS only. Specifies the CICS Operator Class required for each of the eight PCMM Administrative functions.
- Class code 1 controls access to menu function 1, Review Transmission Logs.
- Class code 2 controls access to menu function 2, Profile Maintenance.
- Class code 3 controls access to menu function 3, PC ID Maintenance.
- Class code 4 controls access to menu function 4, Profile Directory.
- Class code 5 controls access to menu function 5, PC ID Directory.
- Class code 6 controls access to menu function 6, Library Directory.
- Class code 7 controls access to menu function 7, (PF7) Options Customization.
- Class code 8 controls access to menu function 7 and 8, User Profile Directory and User Profile Maintenance.

**Note:** When changing Class code 7 (Options Customization), the screen will not allow you to enter a class code that your current CICS signon is not authorized for. This prevents you from locking yourself out of pcMAINFRAME.

**Hardcopy Log Destination:** CICS only. This optional field specifies the name of the DCT destination that the hardcopy log will be written to. This may be a printer, a disk file or another Transient Data destination.

**Log DSN:** TSO only. This is the DSN of the PDS that the log files will be written to. The member names of the logs will be the TSO ID of the user running pcMAINFRAME.

**Security Exit name:** This optional field specifies the PPT name of a user supplied security exit program. This program is invoked by PCMM and PCMX for each menu and transfer request. When a security exit name is entered, that exit is invoked to verify that your current signon is authorized to access this screen. This prevents you from locking yourself out of pcMAINFRAME. Refer to Appendix G - Security Exits in this manual on coding a security exit.

**Activity logging:** This required field controls the logging of transmission activity. Enter either a Y, N, O, or H to indicate the amount of logging you require. Yes indicates both an on-line and hardcopy log will be produced for each transmission. No turns all logging of transmissions off. On-line indicates that only the on-line log will be produced. Hardcopy indicates that only the hardcopy log will be written.

**History logging:** When set to ‘Y’, transmission log records will be keyed with a unique date/timestamp and retained for the interval, in days, entered in the following field. If no interval is coded, log records will be retained for 1 day.

**Note:** Log records maintenance is triggered when a user (PCID) connects to the mainframe for a transfer session. Aged records are purged before the transfer is run.

**Individual form file OK:** TSO only. Indicates whether individual users are permitted to have private form files. If Y is specified, the users may allocate a private form file which contains profiles and file allocation data for private use. Specifying N forces all users to use the system form file.
**Issue Handle Abend:** CICS only. Enter either Yes or No to control the issuing of the CICS Handle Abend condition by the PCMX transmission program. This option allows User exits to get control when an abend occurs. If PCMX gets control during an abend, the user exit will be called with the abend flag set.

**Host Language:** This required field specifies the language that screens and messages used by PCMX and PCMM will be displayed on the mainframe.
1 - English
2 - N/A
3 - French
4 - N/A
5 - Italian
6 - Spanish
7 - Portuguese

**3270 block size:** This required field controls the size of the transmission block being sent to a 3270 type device. To have PCMX use the maximum block size enter 0, otherwise enter 100 to 1920 for Mod 2 terminals or 100 to 3564 for Mod 5 type terminals. Larger block sizes give better throughput, so select 0, zero, unless instructed otherwise.

**Protocol converter block size:** This required field controls the size of the transmission block being sent to a PC attached to a protocol converter. To have PCMX use the maximum block size enter 0, otherwise enter 100 to 1920 for Mod 2 terminals or 100 to 3564 for Mod 5 type terminals.

**Advanced Exit Support:** This required field indicates if advanced user exit options are supported. Advanced exit support requires that all user written exit programs invoked by pcMAINFRAME be able to respond to a "query level" request to set the exit’s function call limits. If this field is "N" (the default), pcMAINFRAME will assume that no advanced exit functions are supported.

**Syncpoint Record Limit:** CICS only. This optional value specifies the number of writes or updates that pcMAINFRAME will do before issuing a CICS Syncpoint. Normally, a syncpoint is performed at the end of each request. With this option set, the syncpoint will also be issued after the specified number of writes or updates during a request. If you are uploading to protected resources, this option may help reduce resource utilization.

**Assume 19xx Cutoff year:** This field has been created specifically to add century to any date-related field that does not carry it. The century designation will be based on the value of this field. Valid values for this field are 00 through 99, and spaces. Any year greater than this cutoff year will be 19xx, and less than or equal will be 20xx. If this field contains spaces, which is the default, we will use current century as the century value.

**Compression Method:** These optional methods are used only when the Advanced Compression Option is available.

Up to four 2-byte options can be specified for upload and download. Each option consists of a 2 character string that defines the compression Method (byte1) and the compression Option (byte2) to be used during file transfer.
The compression selection used determines the amount of additional memory needed for both the mainframe and the PC. Therefore, when a transfer is started, the PC also specifies its selections.

The PC and Mainframe selections for upload and download are compared. The first matching set for each category will be used.

Valid values for Byte 1 are:

1. (no compression)
2. (As defined by the PC)
3. A (Method "A", (lempel-zev))

Valid values for Byte 2 are:

- (Use the default option for this method)
- 1 thru 4 (size of dictionary needed)

Dictionary sizes for Method "A" are

<table>
<thead>
<tr>
<th>Dictionary</th>
<th>Dictionary Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option</td>
<td>Entries</td>
</tr>
<tr>
<td>1</td>
<td>4K</td>
</tr>
<tr>
<td>2*</td>
<td>4K</td>
</tr>
<tr>
<td>3</td>
<td>4K</td>
</tr>
<tr>
<td>4</td>
<td>4K</td>
</tr>
</tbody>
</table>

* 2 is the DEFAULT size

The amount of additional memory required on the mainframe:

**UPLOAD**
- the DECOMPRESSION dictionary size
  (PC will need COMPRESSION size)

**DOWNLOAD**
- the COMPRESSION dictionary size
  (PC will need DECOMPRESSION size)
Chapter 6 - Host Initiation

Host Initiation with CFXINIT

Overview

A pcMAINFRAME transfer session can be started from a batch job running in VSE or MVS. The initiating job has complete control over the transfer definition parameters. Two types of transfer can be initiated:

- **BATCH transfers** - File uploads and downloads that are handled completely by the batch job
- **CONTROLLED transfers** - pcMAINFRAME transfers that are started by the batch job, but actually run under the control of CICS or TSO

**Note:** Only ONE type of transfer can be defined in a single job stream. You can run all batch transfers, or all controlled transfers. You cannot combine the two types of transfers.

For BATCH (bulk) file transfer, fixed, ASCII and binary (transparent) file formats are supported. Sequential tape, sequential disk, VSAM, PDS and managed SAM file formats can be transferred. For CONTROLLED (full function) pcMAINFRAME transfers, the host job is used to define the one or more sections of the setup file to be used during actual file transfer, including transfer variables, startscript (logon), endscript (logoff), and specific transfers.

In this type of transfer, the transfer definition data is sent to the PC (as a .HIT file) after the initial connection, then VTAM releases the 3270 session. The PC then connects to CICS or TSO and runs the transfers as specified by the batch job, just as in a user initiated transfer. **Note:** after the mainframe job completes the initial connection to the PC and releases the 3270 session, the job will either

- **END** - the batch job finishes after triggering the CICS/TSO transfer.
- **WAIT** - the batch job can wait for a return code from the CICS/TSO transfer, that will be posted as the RETURN CODE.

Transfer Startup

A transfer is controlled with JCL and statements coded in a command file. The command file includes CONNECT statements, to define the connection to the PC,
SCRIPT statements, to control additional processing on the PC and UPLOAD / DOWNLOAD parameters, used to define file transfers.

The PC must be ready to receive a transfer from the host. For LU2 (3270 type) connections, MA/NW (or MAIN if DOS) must be running on the PC with an active connection to a VTAM session. For LU6.2 and direct TCP/IP connections, the PC application will be automatically started once the connection is established.

**Note:** the PC setup file must be coded with the ‘wait for host’ or ‘wait for host multiple’ option. This causes the PC application to wait for the host to job to connect to its terminal and initiate the transfer process on the PC.

**VTAM Considerations**

The information necessary to define the host initiation and transfer facility to VTAM is contained in the *pcMAINFRAME Installation Guide*. Refer to that manual for detailed information. VTAM updates are required for LU2 and LU6.2 connections.

**LU6.2 Usage notes**

If you use an LU6.2 connection, your SNA server and VTAM must be configured to support the LU6.2 connection that the host will make to the PC(s). The *pcMAINFRAME Installation Guide – APPENDIX D*, contains detailed Configuration Requirements for LU6.2 connections with pcMAINFRAME Express.

Please contact your Network Administrator or Systems Programmer for further information on the LU6.2 configuration at your facility.

---

**BATCH Data File Transfer**

Host initiation can be used to send or receive one or more mainframe data files to or from a PC. This is called batch file transfer. Once the connection to the PC is made, all transfer and script commands are processed. Return codes can be tested to control execution of command sets. On completion, the connection is dropped and the JOBSTEP is ended.

**Record Length Considerations**

When uploading to variable length files, the length of the records will be the length of the PC file record. Records in fixed format files will be truncated or padded to the specified record size. The default PAD character is spaces, nulls (binary zeros) will be used if the PAD(NULL) parameter is used. When uploading to a VSAM file, the RECORDSIDE(nnn) parameter will force a write for the record length specified. Downloaded records will be written with the length of the mainframe record and be delimited with a CF/LF. ASCII format processing will truncate trailing spaces before writing the PC. To read or write PC file records without a CF/LF delimiter, the PCRECSIZE(nnn) parameter must be used.

**KSDS VSAM Upload Considerations**

pcMAINFRAME uses key processing for KSDS uploads. The entire key must be uploaded. If a record with a matching key exists on the file, the uploaded record will replace it.
Audit Files

You can select one of three different formats to create an optional audit trail file.

Fixed Format - creates a file with a single fixed record for each transfer or script processed. This type of audit file could be used to update a database or VSAM file to provide online query of transfer status.

Basic Format - creates a file with the same data as fixed in comma delimited format. A basic format audit file could be downloaded to a PC, then added to a database to provide transfer status look-up to a centralized PC support group.

Log Format - creates a file with a series of 80-byte log record in display format that details the results of each transfer or script processed. The Log format audit file is an easy to read "transfer report" that can be viewed directly on the mainframe or on a PC after download.

Audit File Data

The following fields are defined in the audit record.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Pos</th>
<th>Len</th>
<th>Data Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUDIT-ID</td>
<td>1</td>
<td>12</td>
<td>Char</td>
<td></td>
</tr>
<tr>
<td>LUNAME</td>
<td>13</td>
<td>08</td>
<td>Char</td>
<td></td>
</tr>
<tr>
<td>APPLID</td>
<td>21</td>
<td>08</td>
<td>Char</td>
<td></td>
</tr>
<tr>
<td>NUMBER</td>
<td>29</td>
<td>03</td>
<td>Num</td>
<td></td>
</tr>
<tr>
<td>RUN-TIMESTAMP</td>
<td>32</td>
<td>19</td>
<td>Char</td>
<td>yyyy-mm-dd-hh.mm.ss</td>
</tr>
<tr>
<td>START-TIMESTAMP</td>
<td>51</td>
<td>19</td>
<td>Char</td>
<td>yyyy-mm-dd-hh.mm.ss</td>
</tr>
<tr>
<td>END-TIMESTAMP</td>
<td>70</td>
<td>19</td>
<td>Char</td>
<td>yyyy-mm-dd-hh.mm.ss</td>
</tr>
<tr>
<td>FUNCTION</td>
<td>89</td>
<td>08</td>
<td>Char</td>
<td>UPLOAD/DOWNLOAD/SCRIPT</td>
</tr>
<tr>
<td>PCFILE</td>
<td>97</td>
<td>06</td>
<td>Char</td>
<td></td>
</tr>
<tr>
<td>PROFILE</td>
<td>203</td>
<td>08</td>
<td>Char</td>
<td></td>
</tr>
<tr>
<td>MF-DDNAME</td>
<td>211</td>
<td>08</td>
<td>Char</td>
<td></td>
</tr>
<tr>
<td>RECORD-COUNT</td>
<td>219</td>
<td>09</td>
<td>Num</td>
<td></td>
</tr>
<tr>
<td>BYTE-COUNT</td>
<td>228</td>
<td>11</td>
<td>Num</td>
<td></td>
</tr>
<tr>
<td>COMPRESS-RATE</td>
<td>239</td>
<td>03</td>
<td>Num</td>
<td>99v9 (as %)</td>
</tr>
<tr>
<td>COMPRESS-TYPE</td>
<td>242</td>
<td>02</td>
<td>Char</td>
<td></td>
</tr>
<tr>
<td>SPEED</td>
<td>244</td>
<td>09</td>
<td>Num</td>
<td>(as Bytes per minute)</td>
</tr>
<tr>
<td>SKIPPED-SW</td>
<td>253</td>
<td>01</td>
<td>Char</td>
<td>Y=skipped</td>
</tr>
<tr>
<td>RUN-SKIP-IND</td>
<td>254</td>
<td>01</td>
<td>Char</td>
<td>R=runif S=skipif</td>
</tr>
<tr>
<td>TEST-RC-IND</td>
<td>255</td>
<td>01</td>
<td>Char</td>
<td>M=max L=last U=user</td>
</tr>
<tr>
<td>TEST-CONDITION</td>
<td>256</td>
<td>02</td>
<td>Char</td>
<td>EQ/GE/LE/NE/LE/LT/GT</td>
</tr>
<tr>
<td>TEST-VALUE</td>
<td>258</td>
<td>04</td>
<td>Num</td>
<td></td>
</tr>
<tr>
<td>MAXRC</td>
<td>262</td>
<td>04</td>
<td>Num</td>
<td></td>
</tr>
<tr>
<td>LASTRC</td>
<td>266</td>
<td>04</td>
<td>Num</td>
<td></td>
</tr>
<tr>
<td>USERRC</td>
<td>270</td>
<td>04</td>
<td>Num</td>
<td></td>
</tr>
<tr>
<td>THISRC</td>
<td>274</td>
<td>04</td>
<td>Num</td>
<td></td>
</tr>
<tr>
<td>ERROR-LIT</td>
<td>278</td>
<td>07</td>
<td>Char</td>
<td>WARNING/ERROR/MESSAGE</td>
</tr>
<tr>
<td>ERROR-MSG</td>
<td>285</td>
<td>80</td>
<td>Char</td>
<td></td>
</tr>
<tr>
<td>Unused</td>
<td>365</td>
<td>36</td>
<td>Char</td>
<td></td>
</tr>
</tbody>
</table>

When an audit file is specified, an AUDITID parameter can also be used to uniquely identify the Job in the audit file. This ID will be moved to the first field of EACH audit record.
A sample COBOL copybook, PCMAUFCB.COB can be found in your source definition library. Sample DB2 table definition and load control statements can also be found in source code members PCMAUFRCR.A and PCMAUFLD.A.

Controlled Transfers

The command file provides a great deal of flexibility for starting pcMAINFRAME CICS or TSO transfers. It works with the PC setup file being used by MAINW. It can be used to trigger the transfers already defined in the setup file, or it can be used to control all aspects of the transfer process, including [STARTSCRIPT] and [ENDSCRIPT] processing.

The sections defined on the mainframe will override any corresponding section defined in the setup file on the PC. After all transfer instructions are sent to the PC (as a .HIT file), then the 3270 session is released, the terminal will return to your VTAM menu (USS screen), or if a TRANSFERAPPL was specified, will logon to the applid specified.

At this point, the PC will begin the standard pcMAINFRAME process:

1. Logon or signon to CICS or TSO - setup file [STARTSCRIPT] section
2. Run transfers - setup file [TRANSFER] sections
3. Logoff - setup file [ENDSCRIPT] section

Remember, during this portion of the transfer, the CFXINIT job has no knowledge of what is happening with the transfer. In fact, if the WAIT parameter was not defined, the CFXINIT job will be finished.

If the WAIT parameter is coded on the mainframe job, then the mainframe job will wait for the PC application to:

4. Reconnect to batch JOB with the completion status

Simple controlled Transfers

Any known, repetitive transfer or series of transfers can be defined as a ‘simple transfer’. The setup file on the PC will be run in its entirety once the mainframe initiates the transfer. Any variables in the [transfer] sections of the file can be overridden by the mainframe job.

The following functions are used when defining simple transfers:

CONNECT

VARIABLE - if the PC setup file contains transfer variables.

Complex controlled Transfers

Complex controlled transfers allow full definition of any or all sections of the PC’s setup file. The mainframe definition will REPLACE the section if it exists on the PC. If any transfer is defined in the mainframe job, then only those transfers defined by the mainframe will be run.

The following functions are used when defining controlled transfers:

UPLOAD
The `UPLOAD` and `DOWNLOAD` functions are keyword driven and should be adequate to define most transfers. The `TRANSFER`, `STARTSCRIPT` and `ENDSCRIPT` functions define replacement sections of the setup file. They must be valid setup file statements as defined in Chapter 3 of the `pcMAINFRAME PC Administrator’s Guide`.

Return Codes

An overall return code is set upon completion of the program. The return code can be set in one of two ways:

1. As a normal result of program processing. Possible return codes are defined below.
2. As directed by the USER in a PC script. This feature gives you control over return code setting when transfer processing is conditional.

Startup/Communications Status

The following return codes define the values set during command file validation and communications startup.

20 Unable to start processing. This error cannot be overridden. It is caused by errors such as syntax errors in the command set, a critical problem with the VTAM interface, a problem with the initial connection to the PC, or an open error on the Command or Transfer Report files.

16 Unable to continue processing. This error cannot be overridden. It is caused by a critical communications problem or a critical problem with the PC during processing. If the PC stops sending/accepting data, or if the communication link fails, this return code will be set.

Controlled Transfer Completion Status

After initiating a pcMAINFRAME session, it is possible to “wait” for the transfer to complete and obtain the completion status of the transfer(s). If wait is coded, the return code will reflect the success of the pcMAINFRAME session.

Batch Transfer Status

The following return code definitions are applicable to batch transfers.

12 One or more transfers failed. This error can be caused by a problem on either the PC or the Mainframe. File open errors, out-of-space conditions, invalid file names or directories will set this return code value.

8 One or more PC scripts failed to execute properly. This error is caused when an error occurs in a control script that is running on the PC.
Syntax errors, runout or timeout errors, user BREAKS and other dialog errors will set this return code value.

4 User issued a QUIT during PCSCRIPT processing, and the QUIT value set was NOT 240 - 255.

other Any RETURN CODE from 0 to 15 can be set during PC script process by issuing a QUIT(nnn) statement where nnn is between 240 and 255. This request will force the step return code value to 0 to 15 (nnn - 240) AND will immediately end all further processing.

Return Code Testing - Batch transfers

The return codes defined above can be tested before any TRANSFER is run, or during the processing of any PCSCRIPT. The test can be used to either RUN or SKIP the process if the condition is true.

Three different Return Code values can be tested in this way:

- **MAXRC** - the highest non-user return code value set during processing. **Note:** this value will usually be 0, 4, 8 or 12 because both 16 and 20 force immediate termination of all processing.

- **LASTRC** - the return code value set during execution of the last TRANSFER or SCRIPT process that ran. A skipped transfer has no effect on LASTRC. The LASTRC value set by a PCSCRIPT can be:
  - 0 (OK)
  - 4 (any QUIT)
  - 8 (SCRIPT process error)

- **USERRC** - the QUIT(nnn) code value set during PCSCRIPT processing when the QUIT code set is between 33 and 239.

**Note:** if a QUIT(nnn) is issued with a value between 240 and 255, all processing ends and the program Return Code will be set to: 0 - 15 (nnn - 240)
Command File

Command Syntax

All commands have this general structure:

```
FUNCTION parameter(s) terminator
```

**FUNCTION** A function defines a specific action to be performed. The command file can contain several functions and their associated parameters. The CONNECT function must be defined once and only once. Separate each function from its parameter(s) with one or more blanks.

**parameter** One or more keyword parameters follow the function. Keyword parameters may be in any order following the verb.

Some keyword parameters require values or names. In this case, the value or name must be enclosed in parenthesis. For example:

```
MEMBER(membername)
```

Separate all parameters with one or more spaces or commas.

**Continuing parameters**

Parameters may be continued over any number of lines, but if a parameter is split between lines the continuation character, "-", must be the last character on the line.

The following example requires a continuation character:

```
UPLOAD
  PC -
    FILE(c:\userdir\ -
        datafile.dat)
```

If a keyword parameter and/or its value is coded on new line, then a continuation is not required.

The following examples do not require a continuation character:

```
UPLOAD
  PROFILE(myprofile)
  PCFILE(datafile.dat)

DOWNLOAD
  PROFILE
    (myprofile)
```
**Terminator**

A terminator indicates the end of a Function Set. It can be one of the following:

- A new FUNCTION
- The end of the command file
- /END

A /END terminator is required at the end of `STARTSCRIPT`, `ENDSCRIPT` and `TRANSFER` data. **Note:** the ‘/’ must begin in column 1.

**Multiple Requests**

Each function requests a different service. All function requests are read and validated before the attempt to connect to PC file is made. A function and all its parameters must be completely coded before the next function can be defined. The following defines number of times any function can be used in a single job.

<table>
<thead>
<tr>
<th>Demand</th>
<th>Limit</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect</td>
<td>Once</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Once</td>
<td>- all responses to transfer variables are defined</td>
</tr>
<tr>
<td>Download/Upload statements</td>
<td>For bulk data transfers, Maximum 100. Requires one or more DDNAME or DLBL/ASSGN in the JOB JCL.</td>
<td></td>
</tr>
<tr>
<td>PCSCRIPT</td>
<td>Maximum 100</td>
<td></td>
</tr>
<tr>
<td>STARTSCRIPT</td>
<td>Once</td>
<td></td>
</tr>
<tr>
<td>ENDSERPT</td>
<td>Once</td>
<td></td>
</tr>
<tr>
<td>Transfer</td>
<td>Up to 100 different transfers can be defined</td>
<td></td>
</tr>
</tbody>
</table>
## Command File Format

<table>
<thead>
<tr>
<th>CONNECT</th>
<th>LUNAME(luname)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LU2 (3270) connections</td>
<td>APPLID(applid)</td>
</tr>
<tr>
<td></td>
<td>[APPLCOUNT(nnn)]</td>
</tr>
<tr>
<td></td>
<td>[APPLPASS(password)]</td>
</tr>
<tr>
<td>LU6.2 connections</td>
<td>LUNAME(sluname)</td>
</tr>
<tr>
<td></td>
<td>APPLID(applid)</td>
</tr>
<tr>
<td></td>
<td>[APPLCOUNT(nnn)]</td>
</tr>
<tr>
<td></td>
<td>[APPLPASS(password)]</td>
</tr>
<tr>
<td>TCP/IP connections</td>
<td>LOGMODE(logmode)]</td>
</tr>
<tr>
<td></td>
<td>[LU62USER(ffpuser)]</td>
</tr>
<tr>
<td></td>
<td>[LU62PASS(ffpswd)]</td>
</tr>
<tr>
<td></td>
<td>[TPN(tpn)]</td>
</tr>
<tr>
<td>All connections</td>
<td>TCPIP(YES)</td>
</tr>
<tr>
<td></td>
<td>HOST(tcp.ip.address)</td>
</tr>
<tr>
<td></td>
<td>PORT(port#)</td>
</tr>
<tr>
<td></td>
<td>[SUBSYSID(subsys)]</td>
</tr>
<tr>
<td></td>
<td>[TCPNAME(MVSregionName)]</td>
</tr>
<tr>
<td></td>
<td>[TPN(tpn)]</td>
</tr>
<tr>
<td>Batch transfers</td>
<td>[RETRY(nnn[,zzz])]</td>
</tr>
<tr>
<td></td>
<td>[NOTRACE</td>
</tr>
<tr>
<td>Controlled transfers</td>
<td>[COMPRESSID(nnnnnnnnn)]</td>
</tr>
<tr>
<td></td>
<td>[DCOPTION(xx)]</td>
</tr>
<tr>
<td></td>
<td>[UCOPTION(xx)]</td>
</tr>
<tr>
<td></td>
<td>[AUDIT(FIXED</td>
</tr>
<tr>
<td></td>
<td>[AUDITID(userdefined)]</td>
</tr>
<tr>
<td>VARIABLE Controlled transfers</td>
<td>[TRANSFERAPPL(applid)]</td>
</tr>
<tr>
<td></td>
<td>[NOWAIT</td>
</tr>
</tbody>
</table>

**VARIABLE**

Controlled transfers  
\#VARNAME(value) 
\#VARNAME(value) .
Figure 6-1: Command File Format

Parameter Descriptions

CONNECT (define connection parameters)

APPLCOUNT(nnn) Optional. The number of APPLs in the pool of APPLs that CFXINIT can use for connect. See the VTAM Considerations section for additional information on APPLCOUNT usage.
APPLID(applid) *Required.* The name of the SNA APPL that CFXINIT will use. See the VTAM Considerations section for additional information on APPLID.

APPLPASS(password) Optional. Password for the APPLID if needed.

AUDIT(NONE | FIXED | BASIC | LOG) *Optional.* Batch transfers only. Used to indicate the type of AUDIT file to be produced. If specified, a //CFAUDIT DD statement or //DLBL is required to define the audit file.

AUDITID(userspecified) *Optional.* Batch transfers only. For audit files, this ID (up to 12 characters) will be inserted in the first field of each audit record produced. If not specified, this field will be null (BASIC) or spaces.

COMPRESSID(nnnnnnn) *Optional.* This ID is supplied by cfSOFTWARE for the duration of an advanced compression option trial. A permanent ID will be supplied for authorized ACO users.

DCOPTION(xx)|UCOPTION(xx) *Optional.* Used to specify one or more compression methods during bulk file download. DCOPTION(***) indicates that the compression option supplied by the PC should be used. For detailed information on advanced compression methods and their effect on memory allocation, see the pcMAINFRAME PC Administrator’s Guide and the Options Customization section in this manual.

HOST(tcp.ip.address) *Required for direct TCP/IP.* The TCP/IP address or DNS name of the remote host (target PC).

LOGMODE *Optional.* For use with LU6.2 connections to define the session logmode. Standard and commonly defined logmodes for LU6.2 are #INTER or #BATCH.

LU62(Y) *Optional.* LU6.2 session indicator. Required for LU6.2 connections. When the connection is made, the PC’s transfer application will be automatically started.

Note: LU6.2 transfers require additional configuration on both the PC and mainframe. See the Installation Guide, Appendix C, for detailed information on LU6.2 configuration.

LU62PASS Optional. For use with LU6.2 connections. Needed only if the PC has LU6.2 security on.

LU62USER Optional. For use with LU6.2 connections. Needed only if the PC has LU6.2 security on.

LUNAME(luname) *Required.* The SNA name of the 3270 terminal (the PC) to which CFXINIT will attempt to connect. For LU6.2 connections, this is the APPC/LU6.2 name assigned to the target PC that is running the LU6.2 stack.

NOTRACE | TRACE *Optional.* Specifies that a trace of the VTAM connect activity be produced. This parameter should be used for troubleshooting only. A TRACE file must be defined in the JCL in order for the trace to be turned on.

NOWAIT | WAIT(nnn) *Optional.* Controlled transfers only. Specifies if the CFXINIT JOB is to wait for the file transfer to complete and obtain the transfer completion status. CFXINIT will wait up to nnn minutes for...
the PC to reconnect and send status information. If the host is not contacted within the WAIT interval, or if the transfer is unsuccessful, the job step will end with a return code of 8. This has no affect during bulk uploads or downloads.

PORT(nnn) Required for direct TCP/IP. The TCP port number to which a transfer is directed. The target PC must have APPX TPN service running and configured to ‘listen’ on the specified port number.

RETRY(nnn[,zzz]) Optional. Number of times,(nnn), and intervals in seconds,(zzz) between attempts, trying to establish the connection of the LUNAME if it is busy. If the interval is not specified, the default is 5 minutes.

SUBSYSID(nn) Optional. VSE only. If running more than one TCP/IP for VSE, the two-character sub-system identifier to use for TCP/IP services.

TCPIP(Y) Optional. TCP/IP connection session indicator. Required for TCP/IP connections. When the connection is made, the PC’s transfer application will be automatically started. The target PC must have APPX TPN services running.

TCPNAME(tcpname) Optional. MVS only. If running more than one TCP/IP for MVS, the Address Space Name to use for TCP/IP services.

TPN(appname) Optional. The name of the transfer application to run on the PC. Only used when multiple TPNs are defined on the target PC. The default TPN is CFXPRESS.

TRANSFERAPPL(applid) Optional. Controlled transfers only. The APPLID of the CICS or TSO that is to be used for the pcMAINFRAME transfer. If this parameter is coded, the terminal acquired will be logged into the specified APPLID after the request data has been sent to the PC, and before the actual file transfers are started.

**Batch File Transfer Parameters**

**DOWNLOAD/UPLOAD**

**SKIPIF(MAXRC | LASTRC | USERRC, NE | GE | GT | EQ | LE | LT, nnn) Optional.** This parameter may be used to conditionally skip the transfer defined in this process. For example, the statement SKIPIF(MAXRC,G T,4) would cause the transfer to be bypassed if ANY prior processes returned a value greater than 4.

**RUNIF(MAXRC | LASTRC | USERRC, NE | GE | GT | EQ | LE | LT, nnn) Optional.** This parameter may be used to run the transfer defined in this process ONLY if the condition is true.

**Note:** A SKIPIF statement takes precedence over a RUNIF statement if both are defined in the same process. The RUNIF is effectively ignored.

**DDNAME Optional.** For MVS, specifies the DDNAME of the file to be transferred.
For VSE, the DLBL name of the file to be transferred. If not specified, pcMAINFRAME uses the default as defined in the JCL requirements section that follows.
SYSNUM(nnn) For VSE ONLY. The logical unit number (SYSnnn) used in the extent statement for the file to be transferred. The defaults are SYS020 for TAPE and SYS030 for DISK.

DEV(DISK | TAPE | ESDS | KSDS) The data file device type. If not specified, the default is DISK. VSE Note: managed SAM files must be defined as ESDS.

BLOCKSIZE(nnnnn)|MAX For VSE ONLY. Required for device type tape or disk and record format FB or VB. For FB, must be a multiple of recordsize or MAX.

RECORDSIZE(nnnnn) For VSE ONLY. Required for device type tape or disk. For V or VB files, the record size must include the four byte record descriptor word (RDW).

RECORDFORMAT(F | FB | V | VB) For VSE ONLY. Required for device type tape or disk.

PCFILE(Drive:\Path\Filename) Required. The name of the PC file to be written. A full path can be specified. Maximum length is 128 bytes.

PCRECSIZE(nnnnnn) Optional. The PC record size where a CR/LF is not present or not to be written. When this parameter is used, no CRLF will be written on downloads and none will be required on uploads. Records of the size specified will be transferred.

On upload, the last record will be padded with spaces or low-values if required. Any CRLF in the record is treated as data unless it occurs at the end of the defined PC record size. Transparent format files are read and written with the specified size records.

REPLACE | NOREPLACE | APPEND Optional. Specifies the action to take if the file already exists on the PC. The default action is REPLACE.

FIXED | ASCII | TRANSPARENT | NOXLATE Optional. Defines the PC file data type.

FIXED - The entire record will be transferred to the PC. EBCDIC to ASCII conversion will take place and a CR/LF delimiter will be written at the end of each record.

ASCII - Trailing spaces will be removed from each record. EBCDIC to ASCII conversion will take place and a CR/LF delimiter will be written at the end of each record.

TRANSPARENT - A binary transfer. The data will be transferred without any translation. No record delimiters are written.

NOXLATE - On download, the entire record will be transferred to the PC with no data conversion. A CR/LF delimiter will be written at the end of each record.

On upload, each record, as delimited by a CRLF, will be transferred to the mainframe. Data will not be translated and the record will be written with a length appropriated to the file’s LRECL and RECFM characteristics; records in fixed length files will be padded or truncated; records in variable length files will be written as received. Note: the CRLF will not be treated as data!
**MIXEDCASE | UPPERCASE | WORDPROCESS** Optional. Defines the text mode for non-transparent transfers.

Mixedcase - All valid text mode characters, including lower case letters will be transferred. Any non-text characters (for example, binary zeros) will be converted to spaces.

Uppercase - All valid text mode characters will be transferred. Lower case letters will be converted to upper case and any non-text character encountered will be converted to spaces.

WordProcess - All valid text mode characters will be transferred. Any non-text mode character encountered (for example, binary zeros) will be transferred with no conversion. Thus a HEX’00’ on the mainframe will be a HEX’00’ on the PC.

**CTRLZ | NOCTRLZ** Optional. Determines whether a CTRL-Z, HEX’1A’, will be written at the end of a PC file during download. Writing the CTRL-Z is the default.

**PAD(SPACE | NULL)** Optional. Defines the PAD character. Padding occurs when uploading to a fixed length sequential file and the PC record is shorter than the fixed record length, or when uploading to a VSAM file and the RECORDSIZE(nnn) option is specified. If a pad character is not specified, SPACES will be used.

PCSCRIPT statements
PCSCRIPTS are dialog scripts that will be sent to and run on the PC when they are encountered in the command file. The script can be used to find, copy, rename files, launch PC applications, and test return codes to control continued processing of the transfer session. The predefined variables, &MAXRC, &LASTRC and &USERRC can be tested at any time during script processing.

PCSCRIPT processing requires that you are running pcMAINFRAME for Windows Version 5.3 or above. See the Dialog Programmers Guide for detailed information on the Dialog control language.

**PCSCRIPT**

**Script Statements**

/END

Note: the /END statement is required at the end of each PCSCRIPT section. It must begin in column 1 of the record.

**Controlled file transfer Parameters**

**VARIABLE** (Supply substitution values for transfer variables)

&VARIABLE(value) Optional. Used to send the substitution values of transfer variables (e.g. &profile or &pcfile) that have been defined in a setup file on the PC. When the VARIABLE function is defined, it is assumed that the defined transfer(s) in a setup file will be run.

Note: if transfer variables are defined in the setup file, then the substitution values must be defined here.

**UPLOAD** (Define parameters for file upload).
PROFILE(filename) Required. The name of the mainframe profile to used for this transfer.

PCFILE(Drive:\Path\Filename) Required. The name of the PC file to be uploaded. A full path can be specified. Maximum length is 128 bytes.

MEMBER(membername) Optional. For library transfers, the library member name if it is to be different from the PC file name. For PDS, the PDS member name if it is not the same as the PC file name.

REPLACE | NOREPLACE | APPEND Optional. For library transfers, defines the desired action if the member already exists.

DOWNLOAD (define parameters for file download)

PROFILE(filename) Required. The name of the mainframe profile to used for this transfer.

PCFILE(Drive:\Path\Filename) Required. The name of the PC file to be uploaded. A full path can be specified. Maximum length is 128 bytes.

MEMBER(membername) Optional. For library transfers, the library member name if it is to be different from the PC file name. For PDS, the PDS member name if it is not the same as the PC file name.

REPLACE | NOREPLACE | APPEND Optional. For library transfers, defines the desired action if the member already exists.

TRANSFER (define a transfer using setup file [transfer] section format)

This function is provided to allow definition of any setup file parameter when specifying a transfer. Normally, the UPLOAD or DOWNLOAD functions will be used. If however, special parameters are required, for example, a password for a LIBRARY member, then this function can be used.

The format of the transfer section is: KEYWORD= PARAMETER. Each couplet must be coded on a single line. The following summarizes the transfer section parameters available. For complete documentation on setup file structure, see Chapter 3 of the pcMAINFRAME PC Administrator’s Guide.

TRANSFERID=Transfer Description Optional. A brief description (up to 16 characters) of the transfer.

FUNCTION=UPLOAD | DOWNLOAD | EXECUTE | DL/EXECUTE Required. Defines the transfer direction.

PCFILE=PC file name Required. Defines the PC file for transfer

REPLACEPC=YES | NO | DATA |APPEND Optional. Defines the action to take if the PC file already exists.

PROFILE=Profilename Required. The name of the profile to use during transfer.

PROPASS= ProfilePassword Optional. If a profile is password protected, the profile password.

LIBPASS=LibPassword Optional. If a library member is password protected, the Library member password.

LIBNOTE=Description Optional. Description of a library member. Will be placed in the LIB header. Used during upload.
**REPLACELIB=**YES | NO | APPEND Optional. Defines the action to take if a library member already exists.

**MEMBER=**MEMBER Optional. Defines a library or PDS member name if the member is not the same as the PC file name.

**RESPOND=**"MatchToKeyword","Answer" Optional. Defines the response to a Profile Variable. This response cannot be overridden by the PC user at transfer execution.

**PROMPT=**"MatchToKeyword","Answer" Optional. Defines the default response to a Profile Variable. This response can be overridden by a PC user at transfer execution.

**/END Required.** The /END delimiter is required at the end of the TRANSFER section. /END must begin in column 1.

**STARTSCRIPT** (define a setup file [STARTSCRIPT] section).

This function is provided to allow definition of a STARTSCRIPT that will run on the PC before file transfer starts. The script can be used to automate 3270 functions needed, such as signon. Any valid DIALOG command can be coded in the script. See the separate Across The Boards® Dialog Programmers Guide for a complete reference to Dialog scripts.

**ENDSCRIPT** (define a setup file [ENDSCRIPT] section).

This function is provided to allow definition of an ENDSCRIPT that will run on the PC after completion of file transfers. The script can be used to automate 3270 functions needed, such as log-off. Any valid DIALOG command can be coded in the script. See the separate Across The Boards® Dialog Programmers Guide for a complete reference to Dialog scripts.

---

**Program Processing**

**Process Reporting**

Two reports may be produced, depending upon the type of transfers taking place. The Control Report is created with each job executing CFXINIT. The Transfer Report is created only if BULK data transfer jobs are executed. These reports will be written to your system’s default SYSLST.

**Control Report** The Control Report recaps the Command File control statements and reports on syntax errors, shows any override parameters in effect and displays connect startup messages.

**Transfer Report** The Transfer Report details connect information and provides transfer statistics and status information for each function requested. This could be used as a permanent record of transactions and their results. Only created on BATCH data transfer jobs.
Sample Reports

The following shows the Control Report, created by both type of Host Initiation transfers.

The Transfer Report, below, is produced only with BATCH data transfer jobs.

Program Execution

MAINW (MAIN) must be running on the target PC before a CFXINIT job is submitted to run (unless you are running with a LU6.2 connection). The target LU should be available to VTAM (at a VTAM menu).
Job Control Language Requirements

**VSE JCL Requirements**

<table>
<thead>
<tr>
<th>File Type</th>
<th>DLBL Default*</th>
<th>Logical Unit #</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command File</td>
<td>-</td>
<td>SYSRD</td>
</tr>
<tr>
<td>Status Report</td>
<td>-</td>
<td>SYSLST</td>
</tr>
<tr>
<td>Audit Report</td>
<td>CFAUDIT</td>
<td>SYS035</td>
</tr>
<tr>
<td>Disk Input</td>
<td>DISKIN</td>
<td>SYS030**</td>
</tr>
<tr>
<td>Tape Input</td>
<td>TAPEIN</td>
<td>SYS020**</td>
</tr>
<tr>
<td>KSDS Input</td>
<td>KSDSIN</td>
<td>-</td>
</tr>
<tr>
<td>ESDS Input</td>
<td>ESDSIN</td>
<td>-</td>
</tr>
<tr>
<td>Disk Output</td>
<td>DISKOUT</td>
<td>SYS030**</td>
</tr>
<tr>
<td>Tape Output</td>
<td>TAPEOUT</td>
<td>SYS020**</td>
</tr>
<tr>
<td>KSDS Output</td>
<td>KSDSOUT</td>
<td>-</td>
</tr>
<tr>
<td>ESDS Output</td>
<td>ESDSOUT</td>
<td>-</td>
</tr>
<tr>
<td>Trace (Optional)</td>
<td>SYSLST</td>
<td>SYS081</td>
</tr>
</tbody>
</table>

**MVS JCL Requirements**

<table>
<thead>
<tr>
<th>File Type</th>
<th>Default DDNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command File</td>
<td>INPUT (any fixed 80 byte file)</td>
</tr>
<tr>
<td>Status Report</td>
<td>REPORT</td>
</tr>
<tr>
<td>Any Data File, In/Out</td>
<td>CFXDATA**</td>
</tr>
<tr>
<td>Audit Report</td>
<td>CFAUDIT ***</td>
</tr>
<tr>
<td>Trace (Optional)</td>
<td>TRCVTM1</td>
</tr>
</tbody>
</table>

* Any DLBL name (VSE) or DDNAME (MVS) can be used for bulk data transfers. If the default is not used, a DDNAME(...) parameter is required to define the file name.

** If the default LU number is not used, a SYSNUM(nnn) parameter is required to define the logical unit number.

*** Requires the following DCB characteristics:

<table>
<thead>
<tr>
<th>Audit Type</th>
<th>RECFM</th>
<th>LRECL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIXED</td>
<td>FB</td>
<td>300</td>
</tr>
<tr>
<td>BASIC</td>
<td>VB</td>
<td>304</td>
</tr>
<tr>
<td>LOG</td>
<td>FB</td>
<td>80</td>
</tr>
</tbody>
</table>

Overrides

Any CONNECT parameter can be overridden by coding  
`PARM='KEYWORD(value)'`

on the EXEC statement. Multiple keywords should be separated with a comma. For example:

`PARM='LUNAME(&LUID),RETRY(5)'`
Examples

VSE Examples - Batch

Example 1 - Batch file transfers from KSDS and Sequential Disk files. A fixed audit file is produced.

```
* $ JOB JNM=INITBULK,CLASS=0
* $ LST CLASS=U
// JOB CFXINIT IS HOST INIT PROGRAM
/* -----------------------------------------------------------------
/* TWO INPUT FILES - KSDS AND SD FILE
/* -----------------------------------------------------------------
// LIBDEF *,SEARCH=PCM.UTIL
/* -----------------------------------------------------------------
// DLBL KSDSIN, 'PCM.KSDSFILE', VSAM, CAT=SYSCT
// DLBL SEQDISK, 'PCM.SEQDATA', 0
// EXTENT SYS030, SYSWK1, 0, 513891, 256
// ASSIGN SYS030, DISK, VOL=USER01, SHR
/*
// DLBL CFAUDIT, 'PCM.AUDIT', 0
// EXTENT SYS035, SYSWK1, 0, 13260, 1000
// ASSIGN SYS035, DISK, VOL=SYSWK1, SHR
// EXEC CFXINIT
CONNECT
   AUDIT(FIXED)
   AUDITID(DAILY-ROSTER)
   ----------------------------- // 3270 connection
   LUNAME(HCF01201)
   APPLID(HIFAPLP)
   ----------------------------- // TCP/IP connection
   TCPIP(YES)
   HOST(tcp.ip.address)
   PORT(2500)
   ----------------------------- // LU6.2 connection
   LU62(YES)
   LUNAME(SLU-NAME)
   APPLID(HIFAP01)

* --------------------------------------- KSDS FILE TRANSFER ------
DOWNLOAD
   DEV(KSDS)
   PCFILE(INITBLK1.KS1)

* --------------------------------------- SD FILE TRANSFER -------
DOWNLOAD
   DEV(DISK)
   DDNAME(SEQDISK)
   BLOCKSIZE(10000) RECORDSIZE(500) RECORDFORMAT(FB)
   PCFILE(INITBLK.SQ1)
/*
/* Figure 6-4: Batch file transfer
```
Example 2 - Runs a PCSCRIPT to verify the input file’s existence, uploads to the mainframe, then deletes the PC file.

```
* $ $ JOB JNM=INITUPLD,CLASS=0
* $ $ JOB INITUPLD - PFM HOSTINIT UPLOAD
// SETPAM LUID=SYS40028
/* --------------------------------------------------------------*
/* PCMAINFRAME - UPLOAD FILE IF AVAILABLE
/* IF UPLOADED AND SUCCESSFUL, DELETE THE PC FILE
/* --------------------------------------------------------------*
// LIBDEF *,SEARCH=CFSOFT.TEST4
// ASSGN SYS081,SYSLST
* VSAM MANAGED SAM (IMPLICIT DEFINE)
// DLBL UPDATE, 'PCM.TEST.SALES', 0, VSAM, RECORDS=5000, RECSIZE=500, CAT=CFSCAT
// EXTENT,CFS001
/*
* EXEC CFXINIT,PARM='LUNAME(&LUID)'
CONNECT
APPLID(HIFAPPL)
PCSCRIPT
OPTION LANGLEVEL(3)
CHDIR "C:\MARKET FILES"
IF EXIST "MIDWEST.SALES.DAT" GOTO GOT_FILE ;DOES FILE EXIST?
:NOFILE
QUIT
:GOT_FILE
END
; FILE EXISTS, (END SETS LASTRC = 0)
/END
UPLOAD
RUNIF(LASTRC,EQ,0)
DEV(DISK)
DDNAME(UPDATA)
BLOCKSIZE(MAX)
RECORDSIZE(500)
RECORDFORMAT(FB)
PCFILE(MIDWEST.SALES.DAT)
PCSCRIPT
OPTION LANGLEVEL(3)
IF NUM &LASTRC NE 0 GOTO NO_DELETE ; EVERYTHING OK?
ON ERROR GOTO NO_DELETE
DELETEFILE 'MIDWEST.SALES.DAT'
END
:NO_DELETE
QUIT
; NOFILE, (QUIT SETS LASTRC = 4)
/END
*/
* $ $ EOJ
```

Figure 6-5: Upload to mainframe if PC File available, then delete PC file

Example 3 - Downloads a file, copies it, adding the current date to its name, then notifies the PC user that the file was downloaded successfully.

```
* $ $ JOB JNM=INITDNLD,CLASS=0
* $ $ LST CLASS=U
// JOB INITDNLD - HOST INIT DOWNLOAD
// SETPAM LUID=SYS40028
/* PCMAINFRAME - DOWNLOAD A FILE,
/* COPY WITH DATE AND NOTIFY USER
/* --------------------------------------------------------------*
// LIBDEF *,SEARCH=CFSOFT.TEST4
// ASSGN SYS081,SYSLST
* VSAM ESDS DOWNLOAD
// DLBL ESDSIN, 'PCM.ESDS.STATS', 6, VSAM, BUFND=6, CAT=CFSCAT
/*
* EXEC CFXINIT,PARM='LUNAME(&LUID)'
CONNECT
APPLID(HIFAPPL)
DOWNLOAD
DEV(ESDS)
PCFILE('C:\PCM_WIN\STATS.DAT')
PCSCRIPT
OPTION LANGLEVEL(3)
IF NUM &LASTRC NE 0 GOTO NO_GO ; EVERYTHING OK?
```

---

104 • Chapter 6 - Host Initiation System Administrator’s Guide
SETUP
ON ERROR GOTO NO_GO
SET #FD1 = @SUBSTR(@DATE, 1, 2) ; GET MMDD FOR FILENAME
SET #FD2 = @SUBSTR(@DATE, 4, 2)
SET #mmdd = #fd1 #fd2
COPY1
CHDIR "C:\MARKET FILES" ; GO TO MARKET DIRECTORY
SET #FNAME = @STRCAT("MIDWEST.STATS.", #MMDD, ".DAT")
COPYFILE "C:\PCM_WIN\STATS.DAT" TO @STRCAT("\PCM_WIN\STATS.DAT", 
#FNAME)
NOTIFY1 ; NOTIFY USER
TIMEOUT(30)
ON TIMEOUT GOTO NO_USER ; IF NO RESPONSE, JUST LEAVE
DLGBOX NEW
DLGBOX line1="Midwest Market File 
DLGBOX line2=@STRCAT("for: ", #mmdd, " Downloaded")
DLGBOX button1="OK"
DLGBOX EXECUTE
NO_USER ; IF NO RESPONSE OR OK
END
NO_GO ; IF UNABLE TO COPY
QUIT(33)
END
*/
* $S EOJ

Figure 6-6: Download a file, with copyfile and notification procedures
Example 4 - Downloads a file, runs an external process to send/receive data from client sites, checks the status of this external process, then uploads any transaction file(s) that were received from clients.

```c
* JOB JNM=INITEXEC,CLASS=0
* SET CLASS=U
// JOB INITEXEC - HOSTINIT DOWNLOAD AND RUN
// SETPARM LUID=SYS40028
* PCMAINFRAME - DOWNLOAD A FILE,
* RUN AN EXTERNAL PROCESS TO SEND PRICES TO CLIENT
* RECEIVE ORDERS
* CHECK STATUS OF SEND/RECEIVE
* UPLOAD TRANSACTION FILE
* LIBDEF *,SEARCH=PCM,LIB
// PCMAINFRAME- DOWNLOAD A FILE,
// RUN AN EXTERNAL PROCESS TO SEND PRICES TO CLIENT
// RECEIVE ORDERS
// CHECK STATUS OF SEND/RECEIVE
// UPLOAD TRANSACTION FILE
// EXEC CFXINIT,PARM='LUNAME(&LUID)'
CONNECT
APPLID(HIFAPPL)
DOWNLOAD
DEV(KSDS)
PCFILE(C:\MARKET FILES\MIDWEST.PRICE.DAT)
* PCSCRIPT
OPTION LANGLEVEL(3)
IF NUM &LASTRC NE 0 GOTO NO_GO ; EVERYTHING OK?
* :SEND_RECV ;RUN THE EXTERNAL SEND/RECEIVE PROCESS
CHDIR "C:\PCPLUS"
RUN "C:\PCPLUS\PCPLUS" /FC:\PCPLUS\DIALSITE.ASP
* ;CHECK RETURN SET IN PRIOR PROCESS
* ;RECORD IS: NN1,NN2 (NN1=RC, NN2=COUNT)
ON FILE_ERROR GOTO BAD_DIAL1
ON END_OF_FILE GOTO BAD_DIAL1
* OPEN (1) "DIALSITE.RC" READ
READ (1) NUM #RC, #FCOUNT
CLOSE (1)
IF NUM #RC > 0 GOTO BAD_DIAL
* :SET_FILES
SET  #FD1 = @SUBSTR(@DATE, 1, 2)     ; GET MMDD FOR FILENAME
SET  #FD2 = @SUBSTR(@DATE, 4, 2)
SET  #MMDD = #FD1 #FD2
CALC #SEQ = 0
CALC #QUITRC = 33 + #FCOUNT ; SET RC FOR QUIT
* :LOOP_COPY ; COPY FILES FOR UPLOAD
CALC #SEQ = #SEQ + 1
IF NUM #SEQ > #FCOUNT THEN
GOTO END_COPY
ENDIF
* COPY FILES TO UPLOAD DIRECTORY, DELETE OLD!
SET OFILE = @STRCAT("C:\PCPLUS\ORDER", #SEQ, ".DAT")
SET NFILE = @STRCAT("C:\MARKET FILES\SITE1.ORDER", #SEQ, ".DAT")
COPYFILE #OFILE TO #NFILE
DELETEFILE #OFILE
GOTO LOOP_COPY
* :BAD_DIAL
QUIT(255) ;SET JOB RC TO 15, NO FURTHER PROCESSING
QUIT(254) ;SET JOB RC TO 14, NO FURTHER PROCESSING
QUIT(253) ;SET JOB RC TO 13, NO FURTHER PROCESSING
* :BAD_DIAL1
QUIT(255) ;SET JOB RC TO 15, NO FURTHER PROCESSING
QUIT(254) ;SET JOB RC TO 14, NO FURTHER PROCESSING
QUIT(253) ;SET JOB RC TO 13, NO FURTHER PROCESSING
END
* UPLOAD
RUNIF(USERRC,GT,33) DEV(ESDS)
```
VSE Examples - Controlled

Example 5 - Simple transfer initiation with transfer variable substitution. The variables define the profile and PC file, as well as start/end browse ranges.

```vse
* $$ JOB JNM=INITXFER,CLASS=0
* $$ LST CLASS=U
// JOB CFXINIT IS A HOST INIT PROGRAM
/* -------------------------------------------------------------
/* TEST CFXINIT - HOST INIT AND VARIABLE PARAMETER
/* -------------------------------------------------------------
// LIBDEF *,SEARCH=PCM.UTIL
// EXEC CFXINIT
CONNECT
LUNAME(HCF01201)
APPLID(HIFAPPL)
TRANSFERAPPL(CICSPROD)
VARIABLE
&PROFILE1(LIB)
&DATASET(PCMTTEST)
&PROFILE2(SALES1)
&RANGEFROM(REC01)
&RANGETO(REC20)
/*
/4
* $$ EOJ
```

**Figure 6-8:** Controlled transfer with variable substitution

Example 6 - Controlled transfer with mainframe WAIT.

```vse
* $$ JOB JNM=INITBULK,CLASS=0
* $$ LST CLASS=U
// JOB CFXINIT IS HOST INIT PROGRAM
/* CONTROLLED PCMAINFRAME TRANSFER FUNCTION -----------------------
// LIBDEF *,SEARCH=PCM.UTIL
// EXEC CFXINIT
CONNECT
LUNAME(HCF01201)
APPLID(HIFAPPL)
TRANSFERAPPL(CICSPROD)
WAIT(15)
* --------------------------------------- UPLOAD
UPLOAD
PROFILE(LIB)
PCFILE(C:\PCM\MYDATA)
MEMBER(MONTH.RPT)
APPEND
* --------------------------------------- DOWNLOAD
DOWNLOAD
PROFILE(SALES1)
PCFILE(C:\PCM\TEST1)
/*
/4
* $$ EOJ
```

**Figure 6-9:** Controlled transfer with WAIT
Example 7 - Controlled transfer with defined scripts and transfer section.

```plaintext
* $J JOB JOB=INITXFER,CLASS=0
* $L JOB=U
* JOB CFXINIT IS A HOST INIT PROGRAM
* --------------------------------------------------------------
// LIBDEF *,SEARCH=FCM.UTIL
// EXEC CFXINIT
CONNECT LUNAME(HCF01201)
APPLID(HIFAPPL)
TRANSFERAPPL(CICSPROD)
STARTSCRIPT
SEND 'B' ENTER
WAIT 'NEW PASSWORD'
WAIT RESPONSE
SEND PF6
WAIT RESPONSE
/END
* --------------------------- ENDSCRIPT TO SIGN OFF CICS
ENDSCRIPT
WAITFOR RESPONSE
SEND RESET CLEAR
WAITFOR RESPONSE
SEND 'CSSF LOGOFF' ENTER
/END
* --------------------------- UPLOAD TO LIB
UPLOAD
PROFILE(LIB)
PCFILE(INITTEST.UP1)
REPLACE
* --------------------------- DOWNLOAD A PASSWORD PROTECTED MEMBER
TRANSFER
TRANSFERID=TEST DOWNLOAD
FUNCTION=DOWNLOAD
PROFILE=LIB
PCFILE=C:\PCM_WIN\INIT.LIB
LIBPASS=MYPASSWD
/END
/ &
* $EOJ
```

**Figure 6-10: Controlled Transfer with setup file sections**

**MVS Examples - Batch**

Example 1 – A Batch file jobstream that downloads a VSAM KSDS and a sequential Disk file. This will produce a fixed audit file.

```plaintext
//JOBNAME JOB (acct)
//*
//BULK FILE TRANSFER 3 FILES (sequential, KSDS, ESDSsds
// #1 - TAKE DEFAULT DDNAME (CFXDATA)
// #2 - TAKE DEFAULT DEV (DATA2)
//*
//JOBLIB DD DSN=PCMAIN.FRAME.LOADLIB,DISP=SHR
//*
//STEP1 EXEC PGM=CFXINIT
//REPORT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSDUMP DD SYSOUT=* 
///CFAUDIT DD DSN=PCM.TEST.AUDIT,DISP=(,CATALG),
//UNIT=SYSDA,SPACE=(TRK,(50,10),RLSK), 
//DCB=(LRECL=400,BLKSIZE=10000,RECFM=FB)
//CFXDATA DD DSN=PCM.TEST.INFILE(TESTDATA),DISP=SHR
//DATA2 DD DSN=PCM.TEST.CFXKSDS,DISP=SHR
//DATA3 DD DSN=PCM.TEST.CFXESDS,DISP=SHR
//INPUT DD *
//CONNECT LUNAME(HCF01200)
//APPLID(HIF0001)
//COMPRESSID(999999)
//DOWNLOAD
//PCFILE(INITBLK1.SD1)
//DOWNLOAD
//DDNAME(DATA2) DEV(KSDS) PCFILE(INITBLK1.KS1)
//UPLOAD
//DDNAME(DATA3) DEV(ESDS) PCFILE(INITBLK1.ES1)
//*
```

**Figure 6-11: Bulk file transfer – MVS**
The following samples are more complex and demonstrate the power of PCSCRIPT statements.

**Example 2** - Runs a PCSCRIPT to verify the input file’s existence, uploads to the mainframe, then deletes the PC file.

```plaintext
//INITUPLD JOB (ACCT)
/* --------------------------------------------------------------*
/* PCMAINFRAME- UPLOAD FILE IF AVAILABLE
/* IF UPLOADED AND SUCCESSFUL, DELETE THE PC FILE
/* --------------------------------------------------------------*
//JOBLIB DD DSN=PCM.LOADLIB
/*
//XPUPDL1 EXEC PGM=CFXINIT,PARM='LUNAME(CF32701)'
//REPORT DD SYSOUT=* 
//CFXDATA DD DSN=PCM.TEST.MIDWEST.SALES,DISP=(,CATLG),
// UNIT=SYSDA,SPACE=(TRK,(20,5),RLSE), 
// DCB=(LRECL=500,BLKSIZE=10000,RECFM=FB)
//INPUT DD *
CONNECT
APPLID(HIF0001)
PCSCRIPT
OPTION LANGLEVEL(3)
CHDIR "C:\MARKET FILES"
IF EXIST "MIDWEST.SALES.DAT" GOTO GOT_FILE ;DOES FILE EXIST?
;NOFILE
QUIT ; NOFILE, (QUIT SETS LASTRC = 4)
:GOT_FILE
END ; FILE EXISTS, (END SETS LASTRC = 0)
/END
UPLOAD
RUNIF(LASTRC,EQ,0)
DEV(DISK)
PCFILE(MIDWEST.SALES.DAT)
PCSCRIPT
OPTION LANGLEVEL(3)
IF NUM &LASTRC NE 0 GOTO NO_DELETE ; EVERYTHING OK?
ON ERROR GOTO NO_DELETE
DELETEFILE 'MIDWEST.SALES.DAT'
END
;NO_DELETE
QUIT ; NOFILE, (QUIT SETS LASTRC = 4)
/END
/*
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
/*
```

**Figure 6-12:** Upload to mainframe if PC File available, then delete PC file
Example 3 - Downloads a file, copies it, adding the current date to its name, then notifies the PC user that the file was downloaded successfully.

```
//INITDNLD JOB (acct)
/* --------------------------------------------------------------*
/*  PCMAINFRAME - DOWNLOAD A FILE,
/*               COPY WITH DATE AND NOTIFY USER
/* --------------------------------------------------------------*
//JOBLIB DD DSN=PCM.LOADLIB
/*
//XPDNLD1 EXEC PGM=CFXINIT,PARM='LUNAME(SYS40028)'
//REPORT DD SYSOUT=*  
//CFXDATA DD DSN=PCM.TEST.STATFILE,DISP=SHR
//INPUT DD *
/*
CONNECT
APPLID(HIFAPPL)
DOWNLOAD
DEV(ESDS)
PCFILE(C:\\PCM_WIN\\STATS.DAT)
*
PCSCRIPT
OPTION LANGLEVEL(3)
IF NUM $LASTRC NE 0 GOTO NO_GO              ; EVERYTHING OK?
:SETUP
ON ERROR GOTO NO_GO
SET #FD1 = @SUBSTR($DATE, 1, 2)            ; GET MMDD FOR FILENAME
SET #FD2 = @SUBSTR($DATE, 4, 2)
SET #mmdd = #fd1 #fd2
:COPY1
CHDIR "C:\\MARKET FILES"                     ; GO TO MARKET DIRECTORY
SET #FNAME = @STRCAT("MIDWEST.STATS.", #mmdd, ".DAT")
COPYFILE "C:\\PCM_WIN\\STATS.DAT" TO @STRCAT("C:\\PCM_WIN\\STATS.DAT", #FNAME)
:NOTIFY1                                      ;NOTIFY USER
TIMEOUT(30)
ON TIMEOUT GOTO NO_USER                     ;IF NO RESPONSE, JUST LEAVE
DLGBOX NEW
DLGBOX line1="Midwest Market File  ", #FNAME
DLGBOX line2=@STRCAT("for: ", #mmdd, " Downloaded")
DLGBOX button1="OK"
DLGBOX EXECUTE
:NO_USER                                    ;IF NO RESPONSE OR OK
END
:NO_GO                                     ;IF UNABLE TO COPY
QUIT(33)
//END
//SYSPRINT DD SYSOUT=*  
//SYSOUT DD SYSOUT=*  */
```

Figure 6-13: Download a file, with copyfile and notification procedures
Example 4 - Downloads a file, runs an external process to send/receive data from client sites, checks the status of this external process, then uploads any transaction file(s) that were received from clients.

```
//INITEXEC JOB (ACCT)
/* -----------------------------------------------*/
/* PCMAINFRAME - DOWNLOAD A FILE,*/
/* COPY WITH DATE AND NOTIFY USER*/
/* -----------------------------------------------*/
//JOBLIB DD DSN=PCM.LOADLIB
/* */
//XPDSNEX EXEC PGM=CFXINIT,PARM='LUNAME(CF32701)'
//REPORT DD SYSOUT=* 
//KSDSIN DD DSN=VSAM.KSDS.PRICE,DISP=SHR
//KSDSDOUT DD DSN=VSAM.KSDS.ORDER,DISP=SHR
//INPUT DD *
CONNECT
APPLID(HIF0001)
DOWNLOAD
DEV(KSDS)
PCFILE(C:\MARKET FILES\MIDWEST.PRICE.DAT)
DNAME(KSDSSIN)
PCSCRIPT
OPTION LANGLEVEL(3)
IF DDN &LASTRC NE 0 GOTO NO_GO ; EVERYTHING OK?
RUN THE EXTERNAL SEND/RECEIVE PROCESS
SEND_RECV
CHDIR "C:\PCPLUS"
/// RUN "C:\PCPLUS\PCPLUS" "/FC:\PCPLUS\DIALSITE.ASP"
; CHECK RETURN FROM PCPLUS
IF RECORD IS: NN1,NN2   (NN1=RC, NN2=COUNT)
ON FILE_ERROR GOTO BAD_DIAL1
ON END_OF_FILE GOTO BAD_DIAL1
*
OPEN (1) "DIALSITE.RC* READ
READ (1) NUM #RC, #FCOUNT
CLOSE (1)
IF NUM #RC  > 0 GOTO BAD_DIAL
*
SET_FILES
*
SET  #FD1 = @SUBSTR(@DATE, 1, 2) ; GET MMDD FOR FILENAME
SET  #FD2 = @SUBSTR(@DATE, 4, 2)
SET  #MMDD = #FD1 #FD2
CALC #SEQ = 0
CALC #QUITRC = 33 + #FCOUNT ; SET RC FOR QUIT
*
LOOP_COPY ; COPY FILES FOR UPLOAD
CALC #SEQ = #SEQ + 1
IF NUM #SEQ > #FCOUNT THEN
GOTO END_COPY
ENDIF
*
COPY FILES TO UPLOAD DIRECTORY, DELETE OLD!
*    OLD NAME, EG: ORDER1.DAT
*    ORDER2.DAT
*    NEW NAME, EG: SITE1.ORDER1.DAT
*    SITE1.ORDER2.DAT
*
SET OFILE = @STRCAT("C:\PCPLUS\ORDER", #SEQ, ".DAT")
SET NFILE = @STRCAT("C:\MARKET FILES\SITE1.ORDER", #SEQ, ".DAT")
COPYFILE #OFILE TO #NFILE
DELETEFILE #OFILE
GOTO LOOP_COPY
*
END_COPY
QUIT(#QUITRC) ; SET USERRC = 33 + NUMBER OF FILES TO Upload
*
BAD_DIAL
QUIT(255) ; SET JOB RC TO 15, NO FURTHER PROCESSING
*
BAD_DIAL1
QUIT(254) ; SET JOB RC TO 14, NO FURTHER PROCESSING
*
NO_GO
QUIT(253) ; SET JOB RC TO 13, NO FURTHER PROCESSING
/END
```
MVS Examples – Controlled

The following are examples of Controlled transfers.

Example 5 - Simple transfer initiation with transfer variable substitution. The variables define the profile and PC file, as well as start/end browse ranges.

```
//JOBNAME JOB (acct)
//**  HOST INITIATION WITH VARIABLE PARAMETER
//**
//JOBLIB DD DSN=PCMAIN.FRAME.LOADLIB,DISP=SHR
//STEP1 EXEC PGM=CFXINIT
//REPORT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//INPUT DD *
CONNECT LUNAME(HCF01200)
APPLID(HIF0001)
TRANSFERAPPL(CICSPRD2)
VARIABLE
&PROFILE1(LIB)
&PCFILE(DAILY.DAT)
/*
/*
```

Figure 6-15: Controlled transfer with variable substitution
Example 6 - Controlled transfer with mainframe WAIT.

```
//JOBNAME JOB (acct)
/**
/* CONTROLLED FUNCTION - WAIT FOR PC
/**
//JOBLIB DD DSN=PCMAIN.FRAME.LOADLIB,DISP=SHR
//STEP1 EXEC PGM=CFXINIT
//REPORT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//INPUT DD *
CONNECT LUNAME(HCF01200)
   APPLID(HIF0001)
   TRANSFERAPPL(CICSPRD3)
   WAIT(15) 
UPLOAD 
PROFILE(PCMT372)
PCFILE(C:\USRDIR\PCMT372.WK1)
DOWNLOAD 
PROFILE(LIB)
PCFILE(MYDATA) 
*/
/**
```

Figure 6-16: Controlled transfer with WAIT

Example 7 - Controlled transfer with defined scripts and transfer section

```
//JOBNAME JOB (acct)
/**
/* INITIATE TRANSFER - WITH START AND END SCRIPTS
/**
//JOBLIB DD DSN=PCMAIN.FRAME.LOADLIB,DISP=SHR
//STEP1 EXEC PGM=CFXINIT
//REPORT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//INPUT DD *
CONNECT LUNAME(HCF01200)
   APPLID(HIF0001)
   WAIT 
   * --------------------------- STARTSCRIPT TO SIGN ON CICS 
   STARTSCRIPT
   SEND 'D' ENTER
   WAIT 'ENTER ACCOUNT'
   SEND 'ACCOUNT' ENTER
   WAIT RESPONSE
   SEND CLEAR
   WAIT CLEAR
   /END
   * --------------------------- ENDSCRIPT TO SIGN OFF CICS 
   ENDSRcript
   WAIT FOR RESPONSE
   SEND RESET CLEAR
   WAIT FOR RESPONSE
   SEND 'CSSF LOGOFF' ENTER
   /END 
   * ------------------------- UPLOAD TO LIBRARY 
   UPLOAD 
   PROFILE(LIB)
   PCFILE(INITTEST.UP1)
   * ------------------------- DOWNLOAD USING TRANSFER SECTION 
   TRANSFER 
   TRANSFERID=TEST DOWNLOAD 
   FUNCTION=DOWNLOAD 
   PROFILE=LIB 
   PCFILE=C:\PCM_MIN\INIT.LIB 
   MEMBER=TESTDATA 
   LIBPASS=MYPASS 
   /END
```
Chapter 7 - Library Processing

Overview

Purpose

Library files provide a facility for storing and keeping as separate members multiple data files in a single VSAM data set. The different members may be of different data types and record formats. Profiles may operate against members of a library in the same fashion as they operate against other files. This gives the flexibility of dynamically creating members as they are uploaded without having to define new datasets to the host.

Libraries are often used for data collection, data distribution, and as a common pool of shared information between PC users. A batch utility program, CFXLIBU is provided with pcMAINFRAME that moves data between library members and other datasets.

Definition

A prototype library file with sample test data is provided with the product. The FCT entry CFXLIB is used to access the system default library. There is no limit to the number of individual library files that can be defined and used. Often, departmental or workgroup libraries are assigned and used to segregate data and optimize performance.

Each PC user is assigned a private and a public library. By default, CFXLIB is used for both. The PCID record is used to specify each user’s library default.

Security

Data in libraries is secured in multiple ways. First, each member is owned by the PC user that created it, can only be overwritten by the owner. Second, passwords can be used to protect data during download. This, standard class code checking is done before any user can access a LIB profile, and finally, only those libraries assigned to a user can be accessed by that user.
Control

Transfers of library members can be controlled in several ways. A ‘HEADER’ record is always created for each member. This header contains control information that can be used to restrict or record transfer activity. For example, a download limit can be set to define the maximum number of times that a member can be downloaded. TimeStamp control can be used to ensure that, if a member or PC file already exists, only a more recent entry will be transferred. Logging can be activated that will track downloads by PCID.

Batch Access

A full function utility for batch retrieval, add, delete and listing of library members is provided with CFXLIBU.

Processing Capabilities

Generic Transfers

When wildcards are specified during an upload to a library, each PC file indicated by the wildcard will be uploaded and stored with a member name that is the same as the PC file name.

If a member already exists in the library and the library is PUBLIC, the library member will be replaced by the uploaded file. If the library is PRIVATE, members that already exist must be owned by the PC ID doing the upload before they will be replaced. Otherwise the transfer will not take place.

When wildcards are specified during a download from a PUBLIC library, each library member indicated by the wildcard will be selected. If the download is from a PRIVATE library, a matching member will be selected only if the member is owned by the PC ID doing the download, and, if password protected, only if the password is given. If the transfer is automated or unattended and the downloaded file already exists on the PC, the PC file will be replaced. If the transfer is manual, the PC user will be able to specify a new PC file name for any selected member that duplicates an existing PC file or append the transferred data to the end of the existing file.

For both upload and download, timestamp controls will be applied if timestamp control is specified in the member header.

Library Time Stamp Control Processing Rules

A library member’s Timestamp Control flag can be set in one of three ways:

During Create or Update via PCMX file transfer, where the settings defined by the UPLOAD profile are used, during Create or Replace via the CFXLIBU utility, where the time stamp control statements are used and via PCMM Library Directory Process where timestamp controls and values can be modified

For Downloads:

1. If the timestamp control is set at M (Mark), the file will always be downloaded, regardless of the control timestamp. The PC file will then be timestamped with the date/time defined in the library control record.
2. When the control is set at Y, if a PC file already exists, its timestamp will be compared to the control timestamp. The file will be downloaded only if the control timestamp is more recent than that of the PC file.

3. Resume will be allowed only if the timestamp on the PC file matches the control timestamp. During generic downloads, a member whose timestamp does not meet the timestamp criteria will be skipped. A log message will be written to indicate the member(s) bypassed.

**For Uploads**

1. When timestamp control is set to *M*ark, it has no affect on uploads.

2. When timestamp control is set to Y, the timestamp of the PC file to be uploaded is compared to the control timestamp. If the PC file create date is more recent than the control date/time, the file will be uploaded.

3. Resume will be allowed only if the timestamp on the PC file matches the control timestamp.

4. During generic uploads, a member whose timestamp does not meet the control criteria will not be uploaded. A log message will be written to indicate the file(s) that are bypassed.

**Library History Log Processing**

The library history log facility can be used to keep track of all downloads of a given library member. When turned on, logging will keep a record of the date, time and USER ID of each successful download.

There are three ways that activate the library LOG HISTORY flag:

1. During a member create or updated via PCMX file transfer, when the upload profile LOG HISTORY flag is set to Y.

2. During create or replace via the CFXLIBU utility, when the HISTORY parameter is defined in the HEADER section of the control file.

3. During PCMM Lib Directory process, with the LOG HISTORY toggle request.

The library utility program command LISTHIST can be used to obtain a full listing of download history dates.

**Note:** this data is normally deleted during any member delete process. If a member is replaced, its history will be deleted.

**Public and Private Libraries**

pcMAINFRAME supports two different types of libraries: Public and Private. Private libraries are used to store and collect information from individual users and Public libraries are typically used for distribution of data to groups of users. There is a single physical format for libraries. The distinction between Public and Private is a logical one determined by the file type specified in the profile that accesses the library. The same data set can be both a Public and a Private library depending on the file type of the profile used when it is accessed.

The different rules for accessing Public and Private libraries are explained below.

1. **DOWNLOAD, PRIVATE**
   - FCT from the PROFILE (not PCID record)
• Any named member
• If PASSWORDED, PASSWORD must be supplied before download, unless the PCID owns the member.
• If WILDCARD, only members owned by PCID/USERID or "ANY" will be downloaded.
• Download limits honored
• No simultaneous downloads

2. DOWNLOAD, PUBLIC
• FCT from the PROFILE (not PCID record)
• Any named member
• No PASSWORD checking
• If WILDCARD, all members that match WILDCARD criteria
• Download limits honored Warning: Limit may be exceeded if simultaneous downloads
• Simultaneous download of the same member

3. UPLOAD, PRIVATE
• FCT from the PRIVATE LIBRARY in PCID record, unless "/*", then from the PROFILE.
• If member already exists in the LIBRARY, then the member must be "ANY" or owned by the PCID/USERID doing the upload.
• PASSWORD is prompted for before write of member
• If WILDCARD, only members owned by the PCID will be deleted and replaced. NOTE: CFX013 should issues an "INTERRUPT" with member not owned by PCID message.

4. UPLOAD, PUBLIC
• FCT from the PUBLIC LIBRARY in PCID record, unless "/*", then from the PROFILE.
• If member already exists in the LIBRARY, the uploaded member will replace the existing member REGARDLESS of the PCID. The PCID of the uploader will be recorded with the new member.
• PASSWORD is not prompted for.
• If WILDCARD, all members matching the WILDCARD spec will be uploaded. If a member already exists in the LIBRARY, the uploaded member will replace the existing member REGARDLESS of the PCID. The PCID of the uploader will be recorded with each member.

5. PRIVATE LIBRARY DIRECTORY
• FCT from the PRIVATE LIBRARY in PCID record, unless "/*", then user will be prompted for the library FCT.
• Only members owned by "ANY" or the PCID will be displayed

6. PUBLIC LIBRARY DIRECTORY
• FCT from the PUBLIC LIBRARY in PCID record. Note: if "/*" is coded, user will not be allowed to supply an FCT, as pcMAINFRAME cannot know if the named file is truly a PUBLIC library
• All members in the library will be displayed
7. PRIVATE LIBRARY DELETE
   • FCT from the PRIVATE LIBRARY in PCID record, unless "*", then
     user will be prompted for the library FCT.
   • Only members owned by the PCID will be deleted

8. PUBLIC LIBRARY DELETE
   • No deletes of PUBLIC LIBRARY members allowed.

Online Processing

Profile Definition - LIB

The screen shown below is a sample LIB profile. No further profile data need be
entered, but any select, summarized and reformat screens can be used if desired.

---

Information for the following fields is pertinent to LIBRARY profiles:

**File Type:** This required entry will be: LIB or PUBL to indicate a public or private
library transfer.

**Note.** For UPLOADS, the File Name specified will be overridden by the library file
names specified in the user’s ID record unless the user’s ID record indicates "*" in
those fields. For LIB and PUBL downloads, the File Name specified in the profile is
always used.

**Download Count Limit:** Defined only when the file type is defined as LIB or
PUBL. This establishes the maximum number of times a member may be
downloaded. For example, this is used to make sure data is not inadvertently
processed more than once. The count may be updated through the Library Directory
screen.

**Timestamp Control:** Defined only when the file type is defined as LIB or PUBL.
This allows you to timestamp the file created by the profile, preventing users from
overwriting PC files more current than the library file, and from overwriting library
files more current than a PC file.
Select Y to timestamp library members when new members are created or when existing members are updated.

Select M to mark library members when new members are created or when existing members are updated.

Select N to eliminate timestamping for library members when members are created or updated. This selection has no impact on downloads.

**Assigning Libraries to PC Users**

The *PC ID Maintenance* screen in CICS or, for TSO, the *User ID Maintenance* as shown below, is used to assign libraries to a PC user.

---

Select Y to timestamp library members when new members are created or when existing members are updated.

Select M to mark library members when new members are created or when existing members are updated.

Select N to eliminate timestamping for library members when members are created or updated. This selection has no impact on downloads.

**Assigning Libraries to PC Users**

The *PC ID Maintenance* screen in CICS or, for TSO, the *User ID Maintenance* as shown below, is used to assign libraries to a PC user.

---

**Private Library:** This is the FCT name of the pcMAINFRAME private library file to be used by this ID. This is an *optional* feature and if you do not want a user to have access to the private library upload feature, specify “NONE” as the private library entry. The default name is “CFXLIB” which is the Library file shipped with pcMAINFRAME. If the library field is an asterisk (*), the PC ID may upload to any library for which it has profile access. Downloads may be done from any library for which a PC ID has profile access, regardless of the library field contents in its PC ID record. Members in private libraries may be password protected to prevent unauthorized access to the member.

**Public Library:** This is the FCT name of the pcMAINFRAME public library file to be used by this PC. This is an *optional* feature and if you do not want a PC to have access to the public library upload feature, specify “NONE” as the public library entry. The default name is “CFXLIB” which is the Library file shipped with pcMAINFRAME. If the library field is an asterisk (*), the PC ID may upload to any
library for which it has profile access. Downloads may be done from any library for which a PC ID has profile access, regardless of the library field contents in its PC ID record. Members in the public library are for general use and may not be password protected.

Library Directory — CICS Only

This function displays a directory of library members that are present on either the default library file `CFXLIB` or other library files as requested. When this function is invoked, a formatted directory screen with no library information on it is displayed. The cursor is positioned in the **File-Name** field. This field is generic, meaning that you may enter a full or partial filename and the display that follows will begin with your entry or the first valid file name following your partial filename entry. The library name field can also be entered. This field defaults to `CFXLIB`, but can be any valid FCT that refers to a pcMAINFRAME library file.

![Figure 7-4: Library Directory - Status Information](image)

![Figure 7-5: Library Directory - Control Information](image)
There are three formats in the library display as shown in the figures above. You may select the control information screen by pressing \G0B/G29/G1C/G0C or the description screen by pressing \G0B/G29/G14/G13/G0C. Press \G0B/G29/G14/G13/G0C or \G0B/G29/G1C/G0C again to return to the status screen. The directory display may be advanced to the next page, if one is present, by pressing the \G0B/G33/G29/G15/G0C key.

To return to the menu by pressing the \G0B/G33/G29/G14/G0C key or the \G0B/G26/G4F/G48/G44/G55/G0C key.

The fields shown in the Library Directory - Status Information screen include:

**File-Name:** This is the one to twelve character name of the library member.

**PC ID:** This is the one to four character name of the PC that created the library member.

**Status:** This indicates what state the library member is in. Possible values of status are:

- **Complete:** an upload or download has been completed
- **In-Xfer:** an upload or download has been started but not completed
- **Create:** in use by batch or CFXLIB.

**Creation Date:** This is the date that the file was added to the library.

**Record Count:** This is the count of data records contained within this library member. The characters N/A may appear for members that were created with versions of pcMAINFRAME prior to 3.02.

**Download Count:** This is the number of times this member has been downloaded since it was created.

**Download Max:** This is the limit of the number of items this member may be downloaded. This may be altered with this screen or set by default from an option on the profile screen. A value of 0 means there is no limit.

**Allow Overlay:** This switch determines whether a member may be overlaid by an upload of the same member. A value of NO will prevent the upload. This may be used to insure that data is kept until it is processed. This value may be set from this screen or defaulted from the profile.

**Data Type:** This is the data type of the file. It may be blank to indicate a comma delimited or text file; or DIF to indicate a data interchange format file; or TRAN to indicate a binary or transparent file.
Additional fields shown by the Library Directory Description screen include:

Descriptive Field: This is a text description supplied by the PC user at the time the library member was created. If the Header Record is missing for a member, an error message will be displayed in this field.

Additional fields shown by the Control Information Screen include:

Time Stamp edit update: You may enter several “T”s to different lines. They will be processed from top to bottom in the same manner as a deletion.

Time Stamp Control: This indicates if the member is subject to timestamp control logic. This field, along with the actual timestamp date/time, can be altered by placing a T in the first byte of the display line (see below).

Time Stamp mm/dd/ccyy and hh:mm:ss: This displays the CONTROL timestamp for file transfer processing. These fields may be altered by placing a T in the first byte of the display line (see below).

History Log: Indicates if History Logging is on or off for this member. This field can be toggled On or OFF by placing an H in the first byte of the member display line.

Online Maintenance Functions

Certain operations can be performed upon library members directly from the directory screen. To make a change to a member, position the cursor on the line containing the member by using the [] key.

To delete an entire member:

Enter [ and press [Enter]. This operation usually takes a few seconds unless the library member is large. In this case it may be more efficient to delete it with the library batch utility CFXLIBU.

To adjust the download maximum:

Type [ and press [Enter] to increase the maximum

Type [ and press [Enter] to decrease the maximum

Type [ and press [Enter] to set the maximum to zero

To Toggle the Allow Overlay:

Type [ and press [Enter]. If the current value is Yes it will be changed to No and vice versa

Library Utility Program

The library utility program, CFXLIBU can be used to add, find, retrieve, delete, replace, append and list members of a pcMAINFRAME library file in batch. You may now also access user defined libraries with this utility. Sequential tape, sequential disk, card, VSAM ESDS and VSAM space managed SAM files are supported. Requests for services are made via a command file, and a status report is generated.
The following functions are provided by the utility program:

- Retrieve a library member
- Add a library member or replace an existing member
- Update header information
- Append records onto the end of an existing member
- Delete an existing library member
- Provide a full or partial directory of library members
- List access history of a member

**Command File**

**Command Syntax**

All commands have this general structure:

```
VERB parameter(s) terminator
```

**VERB** A verb defines the function to be performed. The command file must start with a verb and may be followed by one or more parameters. Separate each verb from its parameter(s) with one or more blanks.

**parameter** One or more keyword parameters follow the verb. Keyword parameters may be in any order following the verb.

Some keyword parameters require values or names. In this case, the value or name must be enclosed in parenthesis. For example:

```
MEMBER(membername)
```

A keyword parameter may also have a set of sub parameters:

```
ENV (DEVI(DISK) RECORDSIZE(80) BLOCKSIZE(8000))
```

In this case, the list of sub parameters must also be enclosed in parenthesis. Separate all parameters and sub parameters with one or more spaces or commas.

**Continuing parameters** Parameters and sub parameters may be continued over any number of lines. If a parameter is split between lines the continuation character, “-”, must be the last character on the line.

The following example requires a continuation character:

```
ADD mem -
    ber(mymember -
      name)
```

These continuations do not require continuation characters:

```
ADD member(mymember)
pclid(mypc)
```

```
ADD member(
    mymembername
  )
```

**terminator** A terminator indicates the end of a command and can be either a new VERB or the end of the command file.
Multiple Requests

Each verb requests a different service. Verbs are processed one at a time as they are encountered in the command file. Thus a verb, and all its parameters, must be completely coded before the next verb can be coded. The following defines the number of times any verb can be used in a single job.

- **Add** One for each unique, valid DEVI
- **Append** One for each unique, valid DEVI
- **Delete** Unlimited
- **Find** Unlimited
- **List** Unlimited
- **ListHist** Unlimited
- **Retrieve** One for each unique, valid DEVO
- **Setup** One for each unique, valid DEVI
- **UpdateHead** Unlimited

Return Codes

Each command sets a return code upon completion. The return code may be tested by subsequent commands to condition their execution. The value of the return codes are:

- 16 Syntax error
- 12 Critical I/O error
- 8 Missing lib header
- 4 Member not found
- 0 Successful completion

Conditional Processing

Commands may be processed conditionally based upon the status returned by the prior command or the highest status returned by any command. The parameters *maxrc* and *runrc* are used to make the command conditional

Command File Format

```
ADD
  member(membername)
  [replace] [noreplace]
  [maxrc(nn.nn)]
  [runrc(nn.nn)]
  [Header]
    ([pcid(pcid)]
      [password(password)]
      [maxdlcnt(nn.nn)]
      [overlay] [nooverlay]
      [history] [nohistory]
      [timestamp(ccyymmdd|hhmmdd)]
      [tscontrol(MARK|YES|NO)]
      [libnote(notes)]
      [libdatatype(type)])
  [Env]
    ([devi(disk|tape|ESDS)]
     [ddname(ddname)]
     [sysnum(nn)]
     [blocksize(nn.nn)]
     [recordsize(nn.nn)]
     [recordformat(F|FB|V|VB)])
```
<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPEND</td>
<td>member(membername) [maxrc(nnnn)] [runrc(nnnn)] [Header]</td>
</tr>
<tr>
<td></td>
<td>([pcid(pcid)]) [password(password)] [maxdlcnt(nnnnn)] [overlay] [history] [timestamp(ccyymmdd</td>
</tr>
<tr>
<td></td>
<td>[Env] ([devi(card</td>
</tr>
<tr>
<td>DELETE</td>
<td>member(membername</td>
</tr>
<tr>
<td>FIND</td>
<td>member(membername</td>
</tr>
<tr>
<td>LIST</td>
<td>member(memberspec</td>
</tr>
<tr>
<td>LISTHIST</td>
<td>member(membername</td>
</tr>
<tr>
<td>RETRIEVE</td>
<td>member(membername</td>
</tr>
<tr>
<td></td>
<td>([devo(disk</td>
</tr>
<tr>
<td>SETUP</td>
<td>LIBDD(DDNAME)</td>
</tr>
<tr>
<td>UPDATEHEAD</td>
<td>member(membername) [maxrc(nnnn)] [runrc(nnnn)] [Header]</td>
</tr>
<tr>
<td></td>
<td>([pcid(pcid)]) [password(password)] [maxdlcnt(nnnnn)] [overlay] [history] [timestamp(ccyymmdd</td>
</tr>
</tbody>
</table>

*Figure 7-7: Command File Format*
Parameter Descriptions

ADD and APPEND

MEMBER(membername) Required. The name of the library member to be added. The length of the name is twelve bytes or less, and the name must be in valid PC filename format.

MAXRC(nnnn) Optional. A 1 to 4 byte numeric value. This command will be bypassed if any prior step had a return code greater than the value coded.

RUNRC(nnnn) Optional. A 1 to 4 byte numeric value. This command will be processed if the prior step had a return code equal to the value coded.

REPLACE | NOREPLACE Optional. If no replace is specified, then an existing member with the same ‘membername’ will not be replaced. This parameter is not valid with APPEND.

HEADER The parameters specified in the header section will be written to the library member’s header record. If any sub parameter is not given, the specified default will be used. Note: during APPEND the header data will NOT be used unless the member does not already exist and the APPEND is treated like an ADD.

PCID(pcid) The ID of the PC user that ‘owns’ the member. If no ID is specified, ANY will be used, and any user will be able to transfer data with no ownership restrictions.

PASSWORD(password) Up to eight bytes of library password. If specified, then the password must be specified before the member can be transferred.

MAXDLCNT(nnn) This count limits the number of times that the member may be downloaded. If not specified, the member may be downloaded any number of times.

OVERLAY | NOOVERLAY Nooverlay specifies that the member cannot be ‘overlaid’ by a pcMAINFRAME upload. Thus a member cannot be accidentally replaced during file transfer.

LIBNOTE(notes) Up to 32 bytes of descriptive information that will be written to the header and displayed during directory list requests.

LIBDATATYPE(type) Used to specify one of two special data types, D (DIF) or T(transparent). Transparent data will not be translated during file transfer. The default data type is FIXED.

ENV The parameters specified in the environment section define the physical attributes of the input or output file to be processed. For MVS only the DEVI parameter is used and only to define ESDS input.

DEVI(DISK | TAPE | ESDS) The input data file device type. If not specified, the default is DISK. Note: managed SAM files must be defined as ESDS.

DDNAME(ddname) Optional. For MVS, specifies the DDNAME of the file to be transferred. For VSE, the DLBL name of the file to be transferred. If not specified, pcMAINFRAME used the default as defined in the JCL requirements section that follows.
SYSNUM(nnnn) *For VSE ONLY.* The logical unit number (SYSnnn) used in the extent statement for the file to be transferred. The defaults are SYS020 for TAPE and SYS030 for DISK.

BLOCKSIZE(nnnnn) *Required* for device type tape or disk and record format FB or VB. For FB, blocksize must be a multiple of the RECSIZE.

RECSIZE(nnnnn) *Required* for device type tape or disk. For V or VB files, the record size must include the four byte record descriptor word (RDW).

RECORDFORMAT(F | FB | V | VB) *Required* for device type tape or disk.

DELETE

MEMBER(membername|generic name) *Required.* The name of the library member to be deleted. The length of the name is twelve bytes or less. Generic names may be used such as hours*.* , *.pay, or time*.

MAXRC(nnnn) *Optional.* A 1 to 4 byte numeric value. This command will be bypassed if any prior step had a return code greater than the value coded.

RUNRC(nnnn) *Optional.* A 1 to 4 byte numeric value. This command will be processed if the prior step had a return code equal to the value coded.

FIND

MEMBER(membername|generic name) *Required.* The name of the library member to be found. The length of the name is twelve bytes or less. Generic names may be used such as hours*.* , *.pay, or time*.

This command does not produce any output, it only sets the return code for use by other commands.

MAXRC(nnnn) *Optional.* A 1 to 4 byte numeric value. This command will be bypassed if any prior step had a return code greater than the value coded.

RUNRC(nnnn) *Optional.* A 1 to 4 byte numeric value. This command will be processed if the prior step had a return code equal to the value coded.

LIST

MEMBER(memberspec|generic name) Defines the starting location for the member directory. If a full directory is desired, use (), with nothing inside the parentheses. If a partial directory is desired, enter a full or partial member name as a pattern. The listing will begin with the first member whose name is equal to or greater than the pattern, and continue from that point. If a generic name such as hours*.* , *.pay, or time*, is used, only members matching the generic name will be listed.

MAXRC(nnnn) *Optional.* A 1 to 4 byte numeric value. This command will be bypassed if any prior step’s return code is greater than the value coded.

RUNRC(nnnn) *Optional.* A 1 to 4 byte numeric value. This command will be processed if the prior step’s return code is equal to the value coded.
LISTHIST

**MEMBER(memberspec|generic name)** Defines the starting location for the member directory. If a full directory is desired, use (), with nothing inside the parentheses. If a partial directory is desired, enter a full or partial member name as a pattern. The listing will begin with the first member whose name is equal to or greater than the pattern, and continue from that point. If a generic name such as `hours*.*`, `*.pay`, or `time*`, is used only members matching the generic name will be listed.

This command produces a listing of download activity for a member. It prints the PC IDs that downloaded the member and the access times.

**MAXRC(nnnn)** *Optional.* A 1 to 4 byte numeric value. This command will be bypassed if any prior step had a return code greater than the value coded.

**RUNRC(nnnn)** *Optional.* A 1 to 4 byte numeric value. This command will be processed if the prior step had a return code equal to the value coded.

RETRIEVE

**MEMBER(membername|generic name)** *Required.* The name of the library member to be retrieved. The length of the name is twelve bytes or less. Generic names may be used such as `hours*.*`, `*.pay`, or `time*`.

**MAXRC(nnnn)** *Optional.* A 1 to 4 byte numeric value. This command will be bypassed if any prior step had a return code greater than the value coded.

**RUNRC(nnnn)** *Optional.* A 1 to 4 byte numeric value. This command will be processed if the prior step had a return code equal to the value coded.

**ENV** The parameters specified in the environment section define the physical attributes of the input or output file to be processed. For MVS only the DEVO parameter is used and only to define ESDS output.

**DEVO(DISK | TAPE | ESDS)** The output data file device type. If not specified, the default is DISK. *Note:* managed SAM files must be defined as ESDS.

**BLOCKSIZE(nnnnn)** *Required* for device type tape or disk and record format FB or VB. For FB, blocksize must be a multiple of the LRECL.

**RECORDSIZE(nnnnn)** *Required* for device type tape or disk. For V or VB files, the record size must be include the four byte record descriptor word (RDW).

**RECORDFORMAT(F | FB | V | VB)** *Required* for device type tape of disk required.

SETUP

**LIBDD(ddname)** *Optional.* The name of the Library file to be used instead of CFXLIB. Used in combination with the // DLBL (VSE) or //OWNLIB DD (MVS), this enables the user to easily change the library name. This must be the first command after the EXEC CFXLIBU.
The Library file specified in the ddname will remain in effect until that step is completed. The default library name will be CFXLIB if no LIBDD is defined.

**UPDATEHEAD**

**MEMBER(membername)** *Required.* The name of the library member. Any information coded in the Header section will be updated in the library header. The length of the name is twelve bytes or less, and the name must be in valid PC filename format.

**MAXRC(nnnn)** *Optional.* A 1 to 4 byte numeric value. This command will be bypassed if any prior step had a return code greater than the value coded.

**RUNRC(nnnn)** *Optional.* A 1 to 4 byte numeric value. This command will be processed if the prior step had a return code equal to the value coded.

**HEADER** The parameters specified in the header section will be written to the library member’s header record. If any sub parameter is not given, the existing value will left intact.

**PCID(pcid)** The ID of the PC user that ‘owns’ the member. If no ID is specified, ANY will be used, and any user will be able to transfer data with no ownership restrictions.

**PASSWORD(password)** Up to eight bytes of library password. If specified, then the password must be specified before the member can be transferred.

**MAXDLCNT(nnn)** This count limits the number of times that the member may be downloaded. If not specified, the member may be downloaded any number of times.

**OVERLAY | NOOVERLAY** Nooverlay specifies that the member cannot be ‘overlaid’ by a pcMAINFRAME upload. Thus a member cannot be accidentally replaced during file transfer.

**LIBNOTE(notes)** Up to 32 bytes of descriptive information that will be written to the header and displayed during directory list requests.

**LIBDATATYPE(type)** Used to specify one of two special data types, D (DIF) or T(transparent). Transparent data will not be translated during file transfer. The default data type is FIXED.
VSE JCL Requirements

<table>
<thead>
<tr>
<th>File Type</th>
<th>DLBL NAME</th>
<th>Logical Unit #</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library</td>
<td>CFXLIB</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>OWNLIB</td>
<td>-</td>
</tr>
<tr>
<td>Command File</td>
<td>-</td>
<td>SYSRDR</td>
</tr>
<tr>
<td>Status Report</td>
<td>-</td>
<td>SYSLST</td>
</tr>
<tr>
<td>Disk Input</td>
<td>DISKIN</td>
<td>SYS030</td>
</tr>
<tr>
<td>Disk Output</td>
<td>DISKOUT</td>
<td>SYS030</td>
</tr>
<tr>
<td>Tape Input</td>
<td>TAPEIN</td>
<td>SYS020</td>
</tr>
<tr>
<td>Tape Output</td>
<td>TAPEOUT</td>
<td>SYS020</td>
</tr>
<tr>
<td>ESDS Input</td>
<td>ESDSIN</td>
<td>-</td>
</tr>
<tr>
<td>ESDS Output</td>
<td>ESDSOUT</td>
<td>-</td>
</tr>
</tbody>
</table>

MVS JCL Requirements

<table>
<thead>
<tr>
<th>File Type</th>
<th>DDNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library (In or Out)</td>
<td>CFXLIB</td>
</tr>
<tr>
<td></td>
<td>OWNLIB</td>
</tr>
<tr>
<td>Command File</td>
<td>INPUT (any fixed 80 byte file)</td>
</tr>
<tr>
<td>Status Report</td>
<td>REPORT</td>
</tr>
<tr>
<td>List Report</td>
<td>LIST (Required with LIST and LISTHIST)</td>
</tr>
<tr>
<td>Any Data File, In/Out</td>
<td>CFXDATA</td>
</tr>
</tbody>
</table>

VSE Examples

This example in Figure 7-8 demonstrates conditional processing, updating header information and deletes member TESTDATA.1 from the library referred to by CFXLIB, then lists a directory of all the remaining members in the library.

```
* $$ JOB JNM=LIST01,CLASS=0
* $$ LST CLASS=U
/ JOB CFXLIBU IS A LIB UTILITY LIST
/* *****************************************************
// LIBDEF *,SEARCH=PCM.UTIL
// DLBL CFXLIB,'PCM.LIBRARY.TEST4',,VSAM,CAT=IJSYSCT
// EXEC CFXLIBU,SIZE=AUTO
FIND MEMBER (SALESOK)
LIST MEMBER (SALESOK*)
MAXRC(7)
UPDATEHEAD MEMBER (SALESOK)
RUNRC(0)
    HEADER ( PCID(NEW1) PASSWORD(SECRET) TIMESTAMP -
    (19950715/031500) TSCONTROL(MARK))
DELETE MEMBER (TESTDATA.1)
* CREATE A LIST OF ALL MEMBERS
LIST MEMBER ()
/*
*/
* $$ EOJ
```

Figure 7-8: Delete and List
The second example, Figure 7-9, adds TESTDATA.1 to the library referred to by CFXLIB. The keyword parameters are spread over several lines, and one, PCID, is broken in the middle, thus requiring a continuation character. A second member is added using card input.

```
* $$ JOB JNM=LOAD01,CLASS=0
  * $$ LST CLASS=U
  // JOB CFXLIBU IS A LIB UTILITY ADD AND APPEND
  /* ******************************************************
  // LIBDEF *,SEARCH=PCM.UTIL
  // DLBL CFXLIB,'PCM.LIBRARY.TEST4',,VSAM,CAT=IJSYSCT
  // DLBL ESDFSIN,'PCMT.ESDSFILE',,VSAM,CAT=IJSYSCT
  // EXEC CFXLIBU,SIZE=AUTO
  ADD MEMBER (TESTDATA.1)
  NOREPLACE
  ENV (DEVI(ESDS))
  HEADER (PC - ID(DEMO))
  PASSWORD (TEST)
  ADD MEMBER (CARDS) NOREPLACE
  HEADER (PCID(SYST) TIMESTAMP(19950630/120000))
  ENV (DEVI(CARD) EOD(+/+) ESC(++))
  DATA
++*/ $$ JOB JNM=LIBPRINT,CLASS=0
  // JOB LIBPRINT
  // EXEC LIBR
  LISTDIR LIB=PRD1
++*/
++*/ $$ EOJ
*/
LIST MEMBER (CARDS)
RUNRC(0)
*/
* $$ EOJ
```

**Figure 7-9: ADDING a Member using an ESDS file as Input**

**Note:** for the above example, an existing member will not be replaced because the NOREPLACE parameter is coded.

In figure 7-10, data is appended to member TESTDATA.1.

```
* $$ JOB JNM=APP01,CLASS=0
  * $$ LST CLASS=U
  // JOB CFXLIBU IS A LIB UTILITY APPEND
  /* ******************************************************
  // LIBDEF *,SEARCH=PCM.UTIL
  // DLBL CFXLIB,'PCMMAIN.LIBRARY.TEST4',,VSAM,CAT=IJSYSCT
  // DLBL DISKIN,'PCM.TEST.LIBDATA'
  // ASSGN SYSS030,DISK,VOL=USER01,SHR
  // EXEC CFXLIBU,SIZE=AUTO
  APPEND MEMBER (TESTDATA.1)
  ENV (DEVI(DISK)
  BLOCKSIZE (8000)
  RECORDSIZE(80)
  RECORDFORMAT(FB))
/*
*/
* $$ EOJ
```

**Figure 7-10: APPEND a member using a sequential disk file as Input**
In figure 7-11, member TESTDATA.1 is retrieved from the library and written to a sequential disk file.

```
* $$ JOB JNM=RET01,CLASS=U
  * $$ LST CLASS=U
  // JOB CFXLIBU IS A LIB UTILITY RETRIEVE
  /* **********************************************
  // LIBDEF *,SEARCH=PCM.UTIL
  // DLBL CFXLIB,'PCMAIN.LIBRARY.TEST4',,VSAM,CAT=IJSYSCT
  // DLBL DISKOUT,'PCM.TEST.LIBDATA'
  // EXTENT SYS030,USER01,1,0,513891,256
  // ASSGN SYS030,DISK,VOL=USER01,SHR
  // EXEC CFXLIBU,SIZE=AUTO
  RETRIEVE MEMBER (TESTDATA.1)
  ENV {
    DEVO(DISK)
    BLOCKSIZE(4096)
    RECORDSIZE(256)
    RECORDFORMAT(VB)
  }
  /*
  /*
  * $$ EOJ
```

**Figure 7-11: RETRIEVE a member and write to a sequential Disk**

In Figure 7-12, Library File OWNLIB is being accessed instead of CFXLIB in order to delete a member.

```
* $$ JOB JNM=LIDUDOT,CLASS=U
  * $$ LST CLASS=U
  // JOB LIBDEL - CF SOFTWARE LIBRARY UTILITY
  /* ***********************************************
  /* DELETE A LIBRARY MEMBER FROM A USER LIBRARY FILE
  /* ***********************************************
  // LIBDEF *,SEARCH=PCM.UTIL
  // DLBL OWNLIB,'PCMAIN.LIBRARY.TEST4',,VSAM,CAT=IJSYSCT
  // EXEC CFXLIBU,SIZE=AUTO
  SETUP LIBDD(OWNLIB)
  DELETE MEMBER(TESTDATA)
  /*
  /*
  * $$ EOJ
```

**Figure 7-12: Using SETUP LIBDD to Delete a Member**
MVS Examples

//JOBNAME JOB (acct)
//*  DELETE TESTDATA.1 THEN LIST MEMBERS REMAINING IN LIBRARY
//STEP1 EXEC PGM=CFXLIBU
//CFXLIB DD DSN=PCM.LIBRARY.TEST4,DISP=SHR
//REPORT DD SYSOUT=*,DCB=BLKSIZE=2660
//LIST DD SYSOUT=*,DCB=BLKSIZE=2660
//INPUT DD *
DELE  TE MEMBER (TESTDATA.1)
* LIST ALL MEMBERS
LIST MEMBER ()
//

Figure 7-13: Delete and List

//JOBNAME JOB (acct)
//*  ADD MEMBER TESTDATA.1 TO LIBRARY DATA SET FROM ESDS INPUT
//STEP1 EXEC PGM=CFXLIBU
//CFXLIB DD DSN=PCM.LIBRARY.TEST4,DISP=SHR
//CFXDATA DD DSN=PCM.PCM.TEST.ESDSFILE,DISP=SHR
//REPORT DD SYSOUT=*,DCB=BLKSIZE=2660
//INPUT DD *
ADD MEMBER (TESTDATA.1)
NOREPLACE
HEADER {
  PCID (DEMO)
  PASSWORD (TEST)   }
ENV  (DEVI (ESDS) )
//

Figure 7-14: Add a Member

//JOBNAME JOB (acct)
//*  APPEND TO THE END OF TESTDATA.1 FROM PCS INPUT
//STEP1 EXEC PGM=CFXLIBU
//CFXLIB DD DSN=PCM.LIBRARY.TEST4,DISP=SHR
//CFXDATA DD DSN=PCM.TEST.PDSIN(TESTDATA),DISP=SHR
//REPORT DD SYSOUT=*,DCB=BLKSIZE=2660
//INPUT DD *
APPEND MEMBER (TESTDATA.1)
//

Figure 7-15: Append

//JOBNAME JOB (acct)
//*  RETRIEVE TESTDATA.1 AND WRITE TO QSAM
//STEP1 EXEC PGM=CFXLIBU
//CFXLIB DD DSN=PCM.LIBRARY,TEST4,DISP=SHR
//REPORT DD SYSOUT=*,DCB=BLKSIZE=2660
//CFXDATA DD DSN=PCM.TEST.NEWDATA,DISP=(,CATLG),
//        UNIT=DISK,Vol=SER=WORK01,SPACE=(TRK(5,1),RLSE),
//        DCB=(LRECL=256,BLKSIZE=4096,RECFM=VB)
//INPUT DD *
RETRIEVE MEMBER (TESTDATA.1)
//

Figure 7-16: Retrieve

//JOBNAME JOB (acct)
//*  DELETE TESTDATA.1 THEN LIST MEMBERS REMAINING IN LIBRARY
//STEP1 EXEC PGM=CFXLIBU
//OWNLIB DD DSN=PCM.LIBRARY.TEST4,DISP=SHR
//REPORT DD SYSOUT=*,DCB=BLKSIZE=2660
//LIST DD SYSOUT=*,DCB=BLKSIZE=2660
//INPUT DD *
SETUP LIDDD(OWNLIB)
DELE  TE MEMBER (TESTDATA.1)
* LIST ALL MEMBERS
LIST MEMBER ()
//

Figure 7-17: Using SETUP LIDDD to Delete and List
This appendix is for pcMAINFRAME users who wish to transfer data directly to and from their Btrieve files. Users of this feature should be familiar with Btrieve data files and must have a valid Btrieve License agreement.

Mainframe Requirements

To upload or download Btrieve files with pcMAINFRAME, a MAINFRAME profile must be created to define the transfer parameters and the mainframe and Btrieve record formats. Use V for the Output Data Format on the Profile Definition Initial Screen.

For detailed information on Profile Definitions, see Chapter 4.

---

**Figure A-1: Profile Definition Initial Screen**

The Special Exit Parameters Screen is used to define the BTRIEVE index to be used during transfer, as well as if the BTRIEVE file can be UPDATED during download, and the OWNER-ID if defined. If an INDEX is defined, it must be non-duplicable.
Each field to be transferred also needs to be defined. Field characteristics (position, length, and data format) must be defined on the Output Format Screen (see Figure A-3, below) for the mainframe file and the BTRIEVE file. The data elements used by the Output Format screen follow the allowable types detailed in Chapter 4 - Using the System.

The specific PC Types, with applicable valid Btrieve data formats are as follows:

<table>
<thead>
<tr>
<th>PC TYPES</th>
<th>BTRIEVE FIELD TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Character or label fields</td>
<td>S String (default)</td>
</tr>
<tr>
<td></td>
<td>SZ ASCIIZ</td>
</tr>
<tr>
<td></td>
<td>SL Length String</td>
</tr>
<tr>
<td>N Numeric fields</td>
<td>N0 Decimal</td>
</tr>
<tr>
<td></td>
<td>N1 Currency</td>
</tr>
<tr>
<td></td>
<td>N5 Integer</td>
</tr>
<tr>
<td></td>
<td>N6 Signed EBCDIC</td>
</tr>
<tr>
<td></td>
<td>N7 Signed ASCII</td>
</tr>
<tr>
<td></td>
<td>N8 Sign Trailing (default)</td>
</tr>
<tr>
<td></td>
<td>N9 Unsigned Binary</td>
</tr>
<tr>
<td>D Date fields</td>
<td>Always Btrieve format</td>
</tr>
<tr>
<td>F Floating point numeric</td>
<td>Always Btrieve FLOAT</td>
</tr>
<tr>
<td>T Current time fields</td>
<td>Always Btrieve time</td>
</tr>
</tbody>
</table>
PC Requirements
The BTRIEVE file must be defined using the BTRIEVE Utility or by an application program that processes BTRIEVE files. The field definitions in the mainframe profile must match the fields that make up the BTRIEVE record. A BTRIEVE data base engine, with release 6.X of the MicroKernel must be available on the PC or network.

Download Processing
The following processing takes place when a BTRIEVE download is executed.

- The BTRIEVE file is OPENED. The file must be defined and it cannot be locked by another user.
- If an INDEX number is defined (in the profile), it is checked to make sure that it is not duplicable. KEY field(s) characteristics are checked and stored.
- If the file is defined as UPDATEABLE (in the profile), an INDEX must be used. If you do not specify the index, 0 (zero) is assumed.
- Records are selected, downloaded and formatted according to the profile definitions.
- Formatted records are “INSERTED” into the BTRIEVE file
  For UPDATEABLE files, the insert may result in a “Duplicate Key”. When this occurs, the downloaded information will UPDATE the existing record.
  If a DUPLICATE KEY error is encountered during INSERT operations to a non-updateable file, the transfer is terminated with a critical error message.
- Records are INSERTED / UPDATED until all records are downloaded.
- The BTRIEVE file is closed.

Upload Processing
The following processing takes place when a BTRIEVE upload is executed.

- The BTRIEVE file is OPENED. The file must be defined and it cannot be locked by another user.
- If an INDEX number is defined (in the profile), it is checked to make sure that it is not duplicable.
- Records are read and formatted for the MAINFRAME.
  If an INDEX is defined, records are read and uploaded in the sequence defined by the specified index.
  If an INDEX is not defined, the records are read and uploaded in ‘physical sequential’ order and may not be in any given sequence.
- Records are read and processed until the end of the file is encountered.
Appendix B - Valid Text Characters

Although text transmission is often the most efficient way to send data, not all of the 256 possible 8-bit characters are valid in text records. If pcMAINFRAME detects any invalid characters in a text record, it replaces them with a space and notifies the user that it has done so.

Valid text characters fall into four categories.

1. Standard text characters
2. Language-specific text characters
3. Characters found in customized text translation tables (optional)
4. Spool control characters (optional)

The tables in this chapter list all characters that are valid for transmission in "BASIC", "DIF", "Fixed", or "Spool" data formats. Note: unless mixed case or word processing mode is selected, lower case characters are translated to upper case.

Use of the text translation tables illustrated in this chapter will guarantee proper translation of special characters that may appear within your applications.

Standard Text Characters

There are 82 standard text characters. They consist of the fifty-two upper and lowercase U.S. alphabetic characters (A-Z), the ten numeric digits (0-9), space, and nineteen special characters.

Although pcMAINFRAME has provisions for altering the ASCII-to-EBCDIC translations associated with other characters (see Customized Text Translation in this chapter), this cannot be done for any of the 82 standard characters.
Standard Character Translation is always as shown in the following table:

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
<th>ASCII</th>
<th>EBCDIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A through I</td>
<td>alphabetic</td>
<td>41-49</td>
<td>C1-C9</td>
</tr>
<tr>
<td>J through R</td>
<td>alphabetic</td>
<td>4A-52</td>
<td>D1-D9</td>
</tr>
<tr>
<td>S through Z</td>
<td>alphabetic</td>
<td>53-5A</td>
<td>E2-E9</td>
</tr>
<tr>
<td>a through z</td>
<td>alphabetic</td>
<td>61-79</td>
<td>B1-B9</td>
</tr>
<tr>
<td>j through r</td>
<td>alphabetic</td>
<td>6A-72</td>
<td>91-9A</td>
</tr>
<tr>
<td>s through z</td>
<td>alphabetic</td>
<td>73-7A</td>
<td>A2-A9</td>
</tr>
<tr>
<td>0 through 9</td>
<td>numeric digits</td>
<td>30-39</td>
<td>F0-F9</td>
</tr>
<tr>
<td>space</td>
<td>space</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>'</td>
<td>double quote</td>
<td>22</td>
<td>7F</td>
</tr>
<tr>
<td>%</td>
<td>percent</td>
<td>25</td>
<td>6C</td>
</tr>
<tr>
<td>&amp;</td>
<td>ampersand</td>
<td>26</td>
<td>50</td>
</tr>
<tr>
<td>'</td>
<td>apostrophe</td>
<td>27</td>
<td>7D</td>
</tr>
<tr>
<td>(</td>
<td>left paren</td>
<td>28</td>
<td>4D</td>
</tr>
<tr>
<td>)</td>
<td>right paren</td>
<td>29</td>
<td>5D</td>
</tr>
<tr>
<td>*</td>
<td>asterisk</td>
<td>2A</td>
<td>5C</td>
</tr>
<tr>
<td>+</td>
<td>plus</td>
<td>2B</td>
<td>4E</td>
</tr>
<tr>
<td>,</td>
<td>comma</td>
<td>2C</td>
<td>6B</td>
</tr>
<tr>
<td>-</td>
<td>dash</td>
<td>2D</td>
<td>60</td>
</tr>
<tr>
<td>.</td>
<td>period</td>
<td>2E</td>
<td>4B</td>
</tr>
<tr>
<td>/</td>
<td>slash</td>
<td>2F</td>
<td>61</td>
</tr>
<tr>
<td>;</td>
<td>semicolon</td>
<td>3B</td>
<td>5E</td>
</tr>
<tr>
<td>:</td>
<td>colon</td>
<td>3A</td>
<td>7A</td>
</tr>
<tr>
<td>&lt;</td>
<td>less-than</td>
<td>3C</td>
<td>4C</td>
</tr>
<tr>
<td>=</td>
<td>equal</td>
<td>3D</td>
<td>7E</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater-than</td>
<td>3E</td>
<td>6E</td>
</tr>
<tr>
<td>?</td>
<td>question</td>
<td>3F</td>
<td>6F</td>
</tr>
<tr>
<td>_</td>
<td>underscore</td>
<td>5F</td>
<td>6D</td>
</tr>
</tbody>
</table>
Language-specific Text Characters

The language-specific text characters consist of 10 to 30 additional special characters; the characters in this group vary depending on the "Controller Language" configuration option of MAINCON. The default is U.S. English.

Preservation of the language-specific characters during transmission is the least reliable aspect of text transmission; these characters are often subject to undesirable translations by protocol converters, older 3270 controllers, incorrectly configured controllers, etc.

To guarantee correct transmission of all language-specific characters, we strongly urge you to use text translation tables (see Customized Text Translation in this chapter). If you are not using custom text tables, keep in mind that:

1. using safe-character mode, either explicitly or via the autodrop facility, provides the U.S. English language-specific set only. Other characters in the language-specific group will be blanked, regardless of the language configured for pcMAINFRAME;
2. the language-specific sets do not necessarily include all the desirable foreign-language characters for the configured language.

If character mistranslation occurs, one of two things will happen:

1. If the mistranslation is a blank character, pcMAINFRAME will fail to connect. pcMAINFRAME will attempt to complete the connection in Safe Character Mode. As mentioned previously, successful connection in Safe Character Mode will preclude non-English special characters unless you are using text translate tables.
2. If the mistranslation is a non-blank character, pcMAINFRAME will connect successfully. The mistranslated data will transfer successfully and no error condition will be reported to either peer application.

The language-specific character groups are defined below.

**U.S. English (controller language = 1 - 3174 Configuration Question #121 = 01)**

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
<th>ASCII</th>
<th>EBCDIC</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>exclamation</td>
<td>21</td>
<td>5A</td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>pound sign</td>
<td>23</td>
<td>7B</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>dollar sign</td>
<td>24</td>
<td>5B</td>
<td></td>
</tr>
<tr>
<td>@</td>
<td>at sign</td>
<td>40</td>
<td>7C</td>
<td></td>
</tr>
<tr>
<td>[</td>
<td>left bracket</td>
<td>5B</td>
<td>BA</td>
<td>2</td>
</tr>
<tr>
<td>\</td>
<td>back slash</td>
<td>5C</td>
<td>E0</td>
<td></td>
</tr>
<tr>
<td>]</td>
<td>right bracket</td>
<td>5D</td>
<td>BB</td>
<td>2</td>
</tr>
<tr>
<td>`</td>
<td>grave accent</td>
<td>60</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td>left brace</td>
<td>7B</td>
<td>C0</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td>broken bar</td>
<td>7C</td>
<td>6A</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td>right brace</td>
<td>7D</td>
<td>D0</td>
<td></td>
</tr>
<tr>
<td>~</td>
<td>tilde</td>
<td>7E</td>
<td>A1</td>
<td></td>
</tr>
</tbody>
</table>
### Dutch (controller language = 2 - 3174 Configuration Question #121 = 47).

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
<th>ASCII</th>
<th>EBCDIC</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>exclamation</td>
<td>21</td>
<td>5A</td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>pound sign</td>
<td>23</td>
<td>7B</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>dollar sign</td>
<td>24</td>
<td>5B</td>
<td></td>
</tr>
<tr>
<td>@</td>
<td>at sign</td>
<td>40</td>
<td>7C</td>
<td></td>
</tr>
<tr>
<td>\</td>
<td>back slash</td>
<td>5C</td>
<td>E0</td>
<td></td>
</tr>
<tr>
<td>]</td>
<td>right bracket</td>
<td>5D</td>
<td>BB</td>
<td>1</td>
</tr>
<tr>
<td>`</td>
<td>grave accent</td>
<td>60</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td>left brace</td>
<td>7B</td>
<td>C0</td>
<td></td>
</tr>
<tr>
<td>]</td>
<td>broken bar</td>
<td>7C</td>
<td>6A</td>
<td></td>
</tr>
<tr>
<td>]</td>
<td>right brace</td>
<td>7D</td>
<td>D0</td>
<td></td>
</tr>
<tr>
<td>~</td>
<td>tilde</td>
<td>7E</td>
<td>A1</td>
<td></td>
</tr>
<tr>
<td>€</td>
<td>cent sign</td>
<td>9B</td>
<td>4A</td>
<td></td>
</tr>
<tr>
<td>¬</td>
<td>logical not</td>
<td>AA</td>
<td>5F</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>vertical bar</td>
<td>B3</td>
<td>4F</td>
</tr>
</tbody>
</table>

### German (controller language = 4 - 3174 Configuration Question #121 = 03).

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
<th>ASCII</th>
<th>EBCDIC</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>§</td>
<td>section sign</td>
<td>15</td>
<td>7C</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td>exclamation</td>
<td>21</td>
<td>4F</td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>pound sign</td>
<td>23</td>
<td>7B</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>dollar sign</td>
<td>24</td>
<td>5B</td>
<td></td>
</tr>
<tr>
<td>^</td>
<td>circumflex</td>
<td>5E</td>
<td>5F</td>
<td></td>
</tr>
<tr>
<td>`</td>
<td>grave accent</td>
<td>60</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>ü</td>
<td>u-umlaut</td>
<td>81</td>
<td>D0</td>
<td></td>
</tr>
<tr>
<td>ä</td>
<td>a-umlaut</td>
<td>84</td>
<td>C0</td>
<td></td>
</tr>
<tr>
<td>Ä</td>
<td>cap a-umlaut</td>
<td>8E</td>
<td>4A</td>
<td></td>
</tr>
<tr>
<td>ö</td>
<td>o-umlaut</td>
<td>94</td>
<td>6A</td>
<td></td>
</tr>
<tr>
<td>Ô</td>
<td>cap o-umlaut</td>
<td>99</td>
<td>E0</td>
<td></td>
</tr>
<tr>
<td>Ü</td>
<td>cap u-umlaut</td>
<td>9A</td>
<td>5A</td>
<td></td>
</tr>
</tbody>
</table>
Italian (controller language = 5 - 3174 Configuration Question #121 = 15).

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
<th>ASCII</th>
<th>EBCDIC</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>§</td>
<td>section sign</td>
<td>15</td>
<td>7C</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td>exclaim mark</td>
<td>21</td>
<td>4F</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>dollar sign</td>
<td>24</td>
<td>5B</td>
<td></td>
</tr>
<tr>
<td>^</td>
<td>circumflex</td>
<td>5E</td>
<td>5F</td>
<td></td>
</tr>
<tr>
<td>é</td>
<td>e-acute</td>
<td>82</td>
<td>5A</td>
<td></td>
</tr>
<tr>
<td>à</td>
<td>a-grave</td>
<td>85</td>
<td>C0</td>
<td></td>
</tr>
<tr>
<td>ç</td>
<td>c-cedilla</td>
<td>87</td>
<td>E0</td>
<td></td>
</tr>
<tr>
<td>è</td>
<td>e-grave</td>
<td>8A</td>
<td>D0</td>
<td></td>
</tr>
<tr>
<td>ì</td>
<td>i-grave</td>
<td>8D</td>
<td>A1</td>
<td></td>
</tr>
<tr>
<td>ò</td>
<td>o-grave</td>
<td>95</td>
<td>6A</td>
<td></td>
</tr>
<tr>
<td>ù</td>
<td>u-grave</td>
<td>97</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>£</td>
<td>pound sterling</td>
<td>9C</td>
<td>7B</td>
<td></td>
</tr>
</tbody>
</table>

Spanish (controller language = 6 - 3174 Configuration Question #121 = 19).

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
<th>ASCII</th>
<th>EBCDIC</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>at sign</td>
<td>40</td>
<td>7C</td>
<td></td>
</tr>
<tr>
<td>[</td>
<td>left bracket</td>
<td>5B</td>
<td>4A</td>
<td></td>
</tr>
<tr>
<td>\</td>
<td>back slash</td>
<td>5C</td>
<td>E0</td>
<td></td>
</tr>
<tr>
<td>]</td>
<td>right bracket</td>
<td>5D</td>
<td>5A</td>
<td></td>
</tr>
<tr>
<td>`</td>
<td>grave accent</td>
<td>60</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td>left brace</td>
<td>7B</td>
<td>C0</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td>right brace</td>
<td>7D</td>
<td>D0</td>
<td></td>
</tr>
<tr>
<td>Pt</td>
<td>Points</td>
<td>9E</td>
<td>B3</td>
<td></td>
</tr>
<tr>
<td>ñ</td>
<td>enya</td>
<td>A4</td>
<td>6A</td>
<td></td>
</tr>
<tr>
<td>ñ</td>
<td>cap-nya</td>
<td>A5</td>
<td>7B</td>
<td></td>
</tr>
<tr>
<td>¬</td>
<td>not sign</td>
<td>AA</td>
<td>5F</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>solid bar</td>
<td>B3</td>
<td>4F</td>
</tr>
</tbody>
</table>
### Brazilian (controller language = 7 - 3174 Configuration Question #121 = 05).

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
<th>ASCII</th>
<th>EBCDIC</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>exclamation</td>
<td>21</td>
<td>4F</td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>pound sign</td>
<td>23</td>
<td>AD</td>
<td>1</td>
</tr>
<tr>
<td>$</td>
<td>dollar sign</td>
<td>24</td>
<td>5A</td>
<td></td>
</tr>
<tr>
<td>@</td>
<td>at sign</td>
<td>40</td>
<td>AC</td>
<td>1</td>
</tr>
<tr>
<td>\</td>
<td>back slash</td>
<td>5C</td>
<td>E0</td>
<td></td>
</tr>
<tr>
<td>]</td>
<td>right bracket</td>
<td>5D</td>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>^</td>
<td>carat</td>
<td>5E</td>
<td>5F</td>
<td></td>
</tr>
<tr>
<td>`</td>
<td>grave accent</td>
<td>60</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td>{</td>
<td>broken bar</td>
<td>7C</td>
<td>CC</td>
<td>1</td>
</tr>
<tr>
<td>}</td>
<td>right brace</td>
<td>7D</td>
<td>90</td>
<td>1</td>
</tr>
<tr>
<td>~</td>
<td>tilde</td>
<td>7E</td>
<td>A1</td>
<td></td>
</tr>
<tr>
<td>Ç</td>
<td>cap C cedilla</td>
<td>80</td>
<td>5B</td>
<td></td>
</tr>
<tr>
<td>é</td>
<td>e-acute</td>
<td>82</td>
<td>D0</td>
<td></td>
</tr>
<tr>
<td>ç</td>
<td>c cedilla</td>
<td>87</td>
<td>5B</td>
<td></td>
</tr>
<tr>
<td>Ê</td>
<td>E-acute</td>
<td>90</td>
<td>4A</td>
<td></td>
</tr>
</tbody>
</table>

### U. K. English (controller language = 8 - 3174 Configuration Question #121 = 22).

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
<th>ASCII</th>
<th>EBCDIC</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>exclamation</td>
<td>2</td>
<td>5A</td>
<td>1</td>
</tr>
<tr>
<td>#</td>
<td>pound sign</td>
<td>23</td>
<td>7B</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>dollar sign</td>
<td>24</td>
<td>4A</td>
<td></td>
</tr>
<tr>
<td>@</td>
<td>at sign</td>
<td>40</td>
<td>7C</td>
<td></td>
</tr>
<tr>
<td>\</td>
<td>back slash</td>
<td>5C</td>
<td>E0</td>
<td></td>
</tr>
<tr>
<td>`</td>
<td>grave accent</td>
<td>60</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td>left brace</td>
<td>7B</td>
<td>C0</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td>broken bar</td>
<td>7C</td>
<td>6A</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td>right brace</td>
<td>7D</td>
<td>D0</td>
<td></td>
</tr>
<tr>
<td>~</td>
<td>tilde</td>
<td>7E</td>
<td>BC</td>
<td>1</td>
</tr>
<tr>
<td>£</td>
<td>pound sterling</td>
<td>9C</td>
<td>5B</td>
<td></td>
</tr>
<tr>
<td>¬</td>
<td>not sign</td>
<td>AA</td>
<td>5F</td>
<td></td>
</tr>
</tbody>
</table>
### French AZERTY (controller language = 3 - 3174
Configuration Question #121 = 30).

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
<th>ASCII</th>
<th>EBCDIC</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>§</td>
<td>section sign</td>
<td>15</td>
<td>5A</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td>exclamation</td>
<td>21</td>
<td>4F</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>dollar sign</td>
<td>24</td>
<td>5B</td>
<td></td>
</tr>
<tr>
<td>†</td>
<td>broken bar</td>
<td>7C</td>
<td>DD</td>
<td>2</td>
</tr>
<tr>
<td>~</td>
<td>tilde</td>
<td>7E</td>
<td>BD</td>
<td></td>
</tr>
<tr>
<td>ü</td>
<td>u-umlaut</td>
<td>81</td>
<td>DC</td>
<td>2</td>
</tr>
<tr>
<td>é</td>
<td>e-acute</td>
<td>82</td>
<td>C0</td>
<td></td>
</tr>
<tr>
<td>â</td>
<td>a-circumflex</td>
<td>83</td>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>ä</td>
<td>a-umlaut</td>
<td>84</td>
<td>43</td>
<td>1</td>
</tr>
<tr>
<td>à</td>
<td>a-grave</td>
<td>85</td>
<td>7C</td>
<td></td>
</tr>
<tr>
<td>ç</td>
<td>c-cedilla</td>
<td>87</td>
<td>E0</td>
<td></td>
</tr>
<tr>
<td>ê</td>
<td>e-circumflex</td>
<td>88</td>
<td>52</td>
<td>1</td>
</tr>
<tr>
<td>ë</td>
<td>e-umlaut</td>
<td>89</td>
<td>53</td>
<td>1</td>
</tr>
<tr>
<td>è</td>
<td>e-grave</td>
<td>8A</td>
<td>D0</td>
<td></td>
</tr>
<tr>
<td>ì</td>
<td>i-umlaut</td>
<td>8B</td>
<td>57</td>
<td>1</td>
</tr>
<tr>
<td>Í</td>
<td>i-circumflex</td>
<td>8C</td>
<td>56</td>
<td>1</td>
</tr>
<tr>
<td>ò</td>
<td>o-circumflex</td>
<td>93</td>
<td>CB</td>
<td>1</td>
</tr>
<tr>
<td>õ</td>
<td>o-umlaut</td>
<td>94</td>
<td>CC</td>
<td>1</td>
</tr>
<tr>
<td>û</td>
<td>u-circumflex</td>
<td>96</td>
<td>DB</td>
<td>2</td>
</tr>
<tr>
<td>ù</td>
<td>u-grave</td>
<td>97</td>
<td>6A</td>
<td>2</td>
</tr>
<tr>
<td>ý</td>
<td>y-umlaut</td>
<td>98</td>
<td>DF</td>
<td>2</td>
</tr>
<tr>
<td>£</td>
<td>pound sterling</td>
<td>9C</td>
<td>7B</td>
<td></td>
</tr>
</tbody>
</table>

### Finnish/Swedish (controller language = S - 3174
Configuration Question #121 = 09).

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
<th>ASCII</th>
<th>EBCDIC</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>exclamation</td>
<td>21</td>
<td>4F</td>
<td></td>
</tr>
<tr>
<td>^</td>
<td>carat</td>
<td>5E</td>
<td>5F</td>
<td></td>
</tr>
<tr>
<td>‼</td>
<td>broken bar</td>
<td>7C</td>
<td>CC</td>
<td></td>
</tr>
<tr>
<td>ü</td>
<td>u-umlaut</td>
<td>81</td>
<td>A1</td>
<td></td>
</tr>
<tr>
<td>é</td>
<td>e-acute</td>
<td>82</td>
<td>E0</td>
<td></td>
</tr>
<tr>
<td>ä</td>
<td>a-umlaut</td>
<td>84</td>
<td>C0</td>
<td></td>
</tr>
<tr>
<td>å</td>
<td>a-circle</td>
<td>86</td>
<td>D0</td>
<td></td>
</tr>
<tr>
<td>Å</td>
<td>cap A-umlaut</td>
<td>8E</td>
<td>7B</td>
<td></td>
</tr>
<tr>
<td>Å</td>
<td>cap A-circle</td>
<td>8F</td>
<td>5B</td>
<td></td>
</tr>
<tr>
<td>ö</td>
<td>o-umlaut</td>
<td>94</td>
<td>6A</td>
<td></td>
</tr>
<tr>
<td>Ô</td>
<td>cap o-umlaut</td>
<td>99</td>
<td>7C</td>
<td></td>
</tr>
</tbody>
</table>
**Hydra (Variation on U.S. English, controller language = H).**

Same as U.S. English, except left and right brace ({} ) are not available.

**Notes**

1. These characters are not in the 3270 national use group for the language. Mistranslation of these characters may cause the pcMAINFRAME connect to fail.
2. These characters are not in the 3270 national use set for the language. Mistranslation of these characters may cause the pcMAINFRAME transfer to fail in older releases of pcMAINFRAME, or the connect to fail in newer releases.

Either of these circumstances can be remedied by using safe-character mode in conjunction with custom text translation tables.

**Customized Text Translation**

At your option, you may provide your own set of ASCII-to-EBCDIC and EBCDIC-to-ASCII translation table for pcMAINFRAME to use. Doing this allows you to:

1. Choose your own translation for any character in the collating sequence except for the standard 82 text characters whose translations are predetermined and cannot be altered.
2. Obtain fail-safe transmission and appropriate translation of all characters in the tables, regardless of the communication link.

To make use of the special text translation described here:

1. Define the translate table using program MAINXLAT. The table you define must be assigned a unique, one-character code (text translate table ID).
2. Using program MAINCON, select "Video Monitor Mode" from the main menu. When that screen is presented, supply the desired table ID in the "Text-Translate Table ID" field. Alternatively, you may pass the ID of the text translate table in the pcMAINFRAME SETUP file (field XLATE-ID). This will override any text translate ID configured through MAINCON.

As a starting point, sample text translation files are included on your pcMAINFRAME PC installation disks. In many cases, these tables can be used without alteration. The sample files are:

- MAINXLAT.TX1 - Text translate for U.S. English
- MAINXLAT.TX2 - Text translate for Dutch
- MAINXLAT.TX3 - Text translate for French AZERTY
- MAINXLAT.TX4 - Text translate for Austrian/German
- MAINXLAT.TX5 - Text translate for Italian
- MAINXLAT.TX6 - Text translate for Spanish
- MAINXLAT.TX7 - Text translate for Brazilian
- MAINXLAT.TX8 - Text translate for U.K. English
- MAINXLAT.TXS - Text translate for Finnish/Swedish
You may use any text translate table with any controller language specification.

**Spool Control Characters**

Spool control characters are supported for text transmission only for spool format. For this format, the following additional characters are acceptable text:

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
<th>ASCII</th>
<th>EBCDIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;CR&gt;</td>
<td>carriage return</td>
<td>0D</td>
<td>0D</td>
</tr>
<tr>
<td>&lt;LF&gt;</td>
<td>line feed</td>
<td>0A</td>
<td>15</td>
</tr>
<tr>
<td>&lt;FF&gt;</td>
<td>form feed</td>
<td>0C</td>
<td>0C</td>
</tr>
<tr>
<td>&lt;CTRL-Z&gt;</td>
<td>end-of-file</td>
<td>1A</td>
<td>1A</td>
</tr>
</tbody>
</table>

The preceding translations for spool control characters take precedence over those defined in user-defined translation tables, but only for spool format; they can still be subject to the translations you specify when you are using other record formats.

The default language assumed by pcMAINFRAME is U.S. English. If this is the appropriate choice for your application, no special configuration is required (although you still may consider providing your own text translation tables).

To configure for any other language, observe the following procedure:

1. If pcMAINFRAME operation for the physical link involves configuring software from other vendors (such as in HLLAPI-based configurations), ensure the software is configured for the appropriate language.
2. Using program MAINCON, select "Video Monitor Mode" from the main menu. When that screen is presented, supply the correct language in the "Controller Language" field.
3. If the communications driver you are using is cfSOFTWARE’s DFT driver, ensure that the appropriate DFT code page file is in a the pcMAINFRAME directory. (file MAINCDx.CPG, where "x" is the language ID).
4. If desired, configure your own text translation tables as described in "Customized Text Translation" earlier in this appendix.

**Performance Considerations**

Safe-character mode and custom text translation are two pcMAINFRAME features which can greatly improve the reliability and functionality of text record transmission. This additional reliability and functionality requires some additional overhead during transmission.

Additional transmission overhead from these features occurs only when sending:

2. Characters in the custom translate tables aside from the standard 82 text characters.

If your data is only sparsely populated with special characters, the protection provided by using one of these methods (especially translate tables) is well worth the cost.
Appendix C - Coding User Exits

Overview

A user exit is an application program that performs functions outside the regular scope of pcMAINFRAME. Exits are written using the CICS command level interface. This appendix describes the rules for coding user exits and provides some samples of their use.

Exits are specified by profiles. When a profile specifies a user exit, the exit is invoked by pcMAINFRAME each time a logical record is ready to be processed. pcMAINFRAME does not transfer data when it first invokes the exit, but informs the exit that files are to be opened and passes it any required parameters. During subsequent invocations, pcMAINFRAME will either pass or request a data record depending on whether the transfer direction is upload or download.

When operating with an exit, pcMAINFRAME continues to perform all functions specified in the profile such as character validation and translation, data reformatting, record selection, breaks, and summarization.

There are special considerations for using user exits with the TSO version of pcMAINFRAME. See the Installation Guide for information on requirements for TSO user exits.

A control block (CFXCOMX) is used to pass control information and data between pcMAINFRAME and the exit in the CICS COMMAREA. User exits may be thought of as I/O routines for pcMAINFRAME.

During an upload with an exit, pcMAINFRAME receives data from the PC, and translates and reformats it according to the profile. However, when it is ready to write a mainframe record, pcMAINFRAME places it in the CFX-RECORD area and does a CICS link to the user exit. The user exit then takes the record from the COMMAREA and performs whatever processing or I/O is required. It then returns it to pcMAINFRAME. Each record is prepared by pcMAINFRAME and passed to the user exit for processing.

Downloads operate in a similar fashion. When pcMAINFRAME would normally read a record, it links to the user exit. The user exit produces the next record, places it in the CFX-RECORD area and returns it to pcMAINFRAME.
filters this record through the selection criteria in the profile and reformats, translates and transfers it to the PC. The user exit is then linked for the next record.

Examples of user exits are distributed in the source library sent with pcMAINFRAME under the names CFXEXIT1 and CFXEXIT2.

CFXEXIT1 accesses multiple VSAM files to produce a single record for downloading to the PC. CFXEXIT2 updates multiple VSAM files from a single record uploaded from the PC.

**Control Block Definition**

Figure C-1 is a COBOL description of the control block used to communicate between the exit and pcMAINFRAME. A machine readable copy of the control block is contained in the source library that was shipped with your system under the name CFXCOMX. The description below identifies the fields in the control block and covers their usage and any special considerations in using them.

**CFX-REQUEST** - this field is filled in by pcMAINFRAME and contains one of the following values:

- **Q** - Query Level. The exit is asked to define the function level of request codes that the exit application can support - **Level 0** (standard processing), **Level 1** (Generic Processing) or **Level 2** (user specified processing). The exit will not be called with any request having a higher level than the level returned in field CFX-FUNCTION-LEVEL. **Note: this call will only be made if the "Advanced Exit Support" option has been enabled.** Otherwise, pcMAINFRAME will assume Level 0. (Refer to the Options and Configuration section in this manual for more information on Advanced Exit Support).

  - **Level 0 Requests:**
    - **O** - Open. The exit is to perform open processing.
    - **P** - Open Resume. The PC has requested an upload restart. The exit should perform open processing and optionally may return the count of the number of records previously uploaded, plus 1, in the field CFX-RESTART-PTR. The PC will resume sending records to the mainframe after it reads the record pointed to by CFX-RESTART-PTR. If the count is not updated, the PC will resume at the first record.
    - **C** - Close. The exit is to perform any required close processing. Control will not be returned to the exit after this request is processed. A close request may come as a result of reaching end-of-file on an upload request, after confirmation from the PC that all records have been received during a download, or by the PC user terminating the request prematurely by depressing [G0B/G28/G56/G46/G0C] on the PC. No messages or queries may be sent during a close request.
    - **R** - Read. The exit should perform the processing required to produce the next logical record.
    - **W** - Write. The exit has been passed a logical record and should perform the processing required to handle it.
Level 1 Requests

- **G** - Generic Open. The PC has requested a generic upload or download with a wildcard file specification. The exit should perform any generic start-up processing.

- **N** - Get Next Generic. The exit should provide the name of the next PC file to be downloaded, or an end-of-file in CFX-RETURN-CODE if there are no more files to be downloaded. A get-next-generic request will follow a generic open, and be re-invoked after standard end-of-file processing for each file in the generic set being downloaded until the exit returns end-of-file, or an error is encountered.

- **U** - Put Next Generic. The exit is provided with the name of the next PC file being uploaded during a generic transfer. No data is sent with this request.

- **L** - Close Generic, all generic processing is complete. The exit should perform any clean-up needed after generic processing. Note: This request will not be issued if the exit has requested a break or flush, or if a close request due to user abort has been issued. No messages or queries may be sent during a generic close request.

Level 2 requests

Any type of processing can be supported by when a User Exit program responds to the "Query Level" request with "Level-2". In this case, the "Function Support Switches" defined below must be set to indicate the functions that are supported by the exit program.

**CFX-REQUEST-MODIFIER** this field is initialized by pcMAINFRAME and will contain an "I" or an "O" to indicate to the open routine whether the exit is to provide input data to pcMAINFRAME or receive output data from pcMAINFRAME.

**CFX-RETURN-CODE** - this field must be set by the exit with one of the following values to indicate the status of exit processing:

- **0** - Zero. Processing during this invocation was successful.

- **B** - Break. Terminate this request and return PC to menu. pcMAINFRAME will not invoke the exit after receiving a break return code so that any closes, dequeues, or other ending processing must be done before returning with a break code. The exit should place a message in CFX-MESSAGE prior to returning describing the reason for the break.

- **F** - Flush. Terminate the request and the session with the PC. pcMAINFRAME will not return to the exit after receiving a flush return code. The exit should place a message in CFX-MESSAGE prior to returning explaining the reason for the flush.

- **M** - Message. pcMAINFRAME will send the message in CFX-MESSAGE to the PC. After the message is sent, the exit will be invoked with the same request as was present when the message was sent. During upload processing, messages may be sent, but
there may be a delay before the message is actually received at the PC. Messages may not be sent during Query Level processing.

- **Q** - Query. pcMAINFRAME will send the message in CFX-MESSAGE to the PC and return the PC’s response in CFX-ANSWER. The length of the answer from the PC will be placed in CFX-ANSWER-LEN. The request will be the same as it was when the query was issued. During upload processing, queries may be sent only during OPEN processing.

- **E** - End of File. This return indicates that all processing for this request is complete. The exit must perform any required dequeues or end processing before issuing this return because pcMAINFRAME will not reinvoke the exit.

- **V** - End of File Verify. This return indicates that all processing for this request is complete, and requests that pcMAINFRAME invoke the exit with a normal close after the PC has verified that all records have been received, or with an abort close if the download is unsuccessful. This return is not valid during generic processing, as the generic close will always signal successful completion of the transfer.

CFX-PC-ID - this is the ID of the PC issuing the request. This is provided by pcMAINFRAME.

CFX-PROFILE-NAME - this field is provided by pcMAINFRAME and contains the name of the profile being processed.

CFX-ENVIRON-ONE - reserved for future use.

CFX-ENVIRON-TWO - reserved for future use.

CFX-SELECT-COUNT - this field is provided by pcMAINFRAME and contains the count (0 - 16) of the number of selection entries specified in the profile and contained in the CFX-SELECT fields.

CFX-SELECT - these fields are provided by pcMAINFRAME and contain the selection criteria specified in the profile. The contents of these selection criteria are provided so that the exit can make use of them if necessary. It is not required that the exit do anything with these entries. pcMAINFRAME will use them for normal processing after the exit returns a data record. However, the exit may use some of the values to reduce the amount of data passed to pcMAINFRAME. From 0 to 16 entries are present as identified in the previous field CFX-SELECT-COUNT. Each entry consists of the following five fields:

CFX-SELECT-NAME - the name of the selection field.

CFX-SELECT-OPER - the operator associated with the selection parameter. It will contain:

2 = Exclude
3 = Include
8 = Start When
9 = Stop After
& = and

See Chapter 3 - Getting Started for a description of the function of these operators.
CFX-SELECT-COMPARE - this is the logical operator that is used for comparison of the data field with the profile value. This may be a “EQ”, “LT”, “GT”, “GE”, “LE”, “NE”, “LO”, or “HI” as described in Chapter 4 - Using the System.

CFX-SELECT-VALUE - this is the value against which the data value in the record will be compared. This definition is used for character values, the next field is used for numeric values.

CFX-SELECT-VALNUM - this is the numeric value against which the data value will be compared. This contains the value in a normalized format of packed length 10.

CFX-PARM-COUNT - this is the count (0 - 16) of the number of parameters that are being passed by pcMAINFRAME to the user exit.

CFX-PARMS - this field is provided by pcMAINFRAME and contains the number of parameter sets identified in the previous field, CFX-PARM-COUNT. Each set contains fields which are similar to the CFX-SELECT fields just described, except that these fields are provided for the exclusive use of the user exit. If any of the fields are defined as variable parameters, those values will have been prompted for and edited by the time the exit is invoked.

CFX-MESSAGE - this area is initialized by the exit and contains messages and attributes of the messages to be sent to the PC as a result of a return code of either "M" or "Q". The message area consists of the following fields:

CFX-ANSWER-TYPE - this field is provided by the exit and contains an attribute specifying the type of processing to be performed on a query answer. The field must contain one of the following valid attributes:

- A accept any answer.
- D answer dark, do not display the answer being entered on the PC screen.
- N answer numeric, accept only a numeric answer from the PC.
- O answer old, this will pre-load a response for the PC user so that if what is pre-loaded is the desired response, the PC user may simply press Return. If the pre-loaded response is not the desired response, the PC user may key over the response before pressing Return. The default response must be placed in CFX-ANSWER before using this answer type.

CFX-MSG1,2,3 - these are 1 - 3 lines of message to be sent to the PC. pcMAINFRAME clears this area to blanks before invoking the exit. The exit must place the message to be sent into these fields. pcMAINFRAME will terminate the message at the first blank line so messages must be loaded sequentially into these fields.

CFX-LANGUAGE - this is the language code specified by the PC in its setup file. This field is provided by pcMAINFRAME and may be used by the exit to supply alternative messages in different languages.
CFX-RESTART-PTR - this field is provided by the exit if open-restart was specified. The count placed in this field is record number that the PC should resume sending on (the number of records already received for the requested profile, PLUS 1). The PC will bypass records on the PC file up to restart-ptr, and will commence sending at that point. The exit is unaware of resume (restart) on download because all required processing is performed by pcMAINFRAME. During an upload, the user exit may force resume processing by putting the appropriate value in CFX-RESTART-PTR during OPEN request processing.

CFX-WP-TYPE-DATA - this field is provided by pcMAINFRAME and contains a flag that indicates whether the profile specifies upper case data, mixed case data, or word processing data. The value of this field is “Y” to indicate that it is word processing data, “U” to indicate upper case data, or “M” to indicate mixed case data.

CFX-TYPE - this field indicates the data type specified in the profile. It may contain any of the following values: A - ASCII data type, B - BASIC data type, D - DIF data type, F - FIXED data type, I - INCLUDE data type, L - Excel data type, T - TRANSPARENT data type, S - SPOOL data type, W - Worksheet, or X - xBASE data type.

CFX-CONNECT - this field is provided by pcMAINFRAME and contains a value indicating the terminal type of the PC connected to the session. It will contain a “3” for 3270 type terminals.

CFX-USER-ABORT-SW - this field is provided by pcMAINFRAME and contains a “Y” if the PC user has aborted the session, or if the transfer was interrupted or terminated unexpectedly. This field will be set before a close request. If on, the exit should perform any cleanup or termination processing that may be required as pcMAINFRAME will not return to the exit.

CFX-PC-MSSG-NUM - this field is supplied by the user exit and may contain a valid pcMAINFRAME message number to be sent to the PC. This is normally used internally by pcMAINFRAME and should not be set by a user-written exit.

CFX-PC-Filename - this pcMAINFRAME supplied field contains the name of the PC file that is being used with this exit. It is either the name of the file from which data is being uploaded or the name of the file to which data is being downloaded.

CFX GENERIC MASK - this pcMAINFRAME supplied field contains a mask representing the wildcard file specification used to initiate a generic transfer.

CFX GENERIC MASK LEFT - this pcMAINFRAME supplied field contains the name portion of the generic file specification, with each wild-chard character represented by low-values.

CFX GENERIC MASK RIGHT - this pcMAINFRAME supplied field contains the extension portion of the generic file specification, with each wild-chard character represented by low-values.

CFX-GML-MAX-POSITION - this pcMAINFRAME supplied field contains a number that points to the last significant byte of the name portion of the wildcard specification.
CFX-GMR-MAX-POSITION - this pcMAINFRAME supplied field contains a number that points to the last significant byte of the extension portion of the wildcard specification.

CFX-GML-COUNT - this pcMAINFRAME supplied field contains a count of the total number of characters (including wildcard characters) in the name portion of the wildcard specification.

CFX-GMR-COUNT - this pcMAINFRAME supplied field contains a count of the total number of characters (including wildcard characters) in the extension portion of the wildcard specification.

CFX-RECORD-PTR - this field is used only if an application is at function level "2" and the "RECORD-PTR-OK-SW" is turned on. The pointer is a fullword pointer to the record to be processed. For upload, pcMAINFRAME will value the pointer before a Write request. For download, the Exit Program must value the pointer after a READ request. For record sizes that exceed 4K bytes, the record pointer option must be used.

CFX-FUNCTION-SWITCHES - The switches in this group must be set by the user exit program when a "Query Function Level" request is made AND the response is Level-2. The switches then indicate the type of processing that the user exit program supports.

CFX-LEVEL-2-SW - Set to "Y" when level-2 switches are to be used.

CFX GENERIC-OK-SW - Set to "Y" if generic requests are supported

CFX-RECORD-PTR-OK-SW - Set to "Y" if the record pointer option is to be used for all READ and WRITE requests.

CFX-ANSWER-LEN - this field is provided by pcMAINFRAME and contains the length of the answer to a query entered from the PC.

CFX-ANSWER - this field is provided by pcMAINFRAME and contains the answer to a query entered by the PC. pcMAINFRAME will set this field to low values if there is no answer requested from the PC.

CFX-RECORD-LEN - this field contains the length of the record passed between the exit and pcMAINFRAME. For downloads, the user exit must place the length of the record being passed to pcMAINFRAME in this field. For uploads, pcMAINFRAME will supply the length of the uploaded record in this field.

CFX-RECORD - this area must contain the logical data record passed between the exit and pcMAINFRAME. For upload processing, pcMAINFRAME will place the next logical record into this area prior to invoking the exit. For download processing, the exit must place the record to be passed to pcMAINFRAME into this area prior to returning.

CFX-WORK - this is an area provided by pcMAINFRAME in which the user exit may store any variables required from one invocation to the next. Since the exit is a "linked" program in CICS, all variables in the program will be reinitialized for each invocation. CFX-WORK is an area in which values, pointers, and other context may be maintained between invocations. This area is initialized the first time an exit is invoked in any pcMAINFRAME session. The area is maintained throughout the pcMAINFRAME session so that exits invoked for different transfer requests may pass data in this area.
Appendix C - Coding User Exits System Administrator's Guide

02 C-CFXCOMX.
05 CFX-REQUEST PIC X.
   88 CFX-OPEN VALUE 'O'.
   88 CFX-OPEN-RESTART VALUE 'P'.
   88 CFX-CLOSE VALUE 'C'.
   88 CFX-READ VALUE 'R'.
   88 CFX-WRITE VALUE 'W'.
   88 CFX-GETNEXT-GENERIC VALUE 'N'.
   88 CFX-PUTNEXT-GENERIC VALUE 'U'.
   88 CFX-OPEN-GENERIC VALUE 'G'.
   88 CFX-CLOSE-GENERIC VALUE 'L'.
   88 CFX-REQUEST-IS-LEVEL-0 VALUES 'O' 'P' 'C' 'R' 'W' 'Q'.
   88 CFX-REQUEST-IS-GENERIC VALUES 'N' 'U' 'G' 'L'.
   05 CFX-REQUEST-MODIFIER PIC X.
   88 CFX-INPUT VALUE 'I'.
   88 CFX-OUTPUT VALUE 'O'.
   05 CFX-RETURN-CODE PIC X.
   88 CFX-OK VALUE '0'.
   88 CFX-BREAK VALUE 'B'.
   88 CFX-FLUSH VALUE 'F'.
   88 CFX-SEND-MSG VALUE 'M'.
   88 CFX-QUERY VALUE 'Q'.
   88 CFX-END-OF-FILE VALUE 'E'.
   88 CFX-EOF-VERIFY VALUE 'V'.
   05 CFX-PC-ID PIC X(4).
   05 CFX-PROFILE-NAME PIC X(16).
   05 CFX-ENVIRON-ONE PIC X.
   05 CFX-ENVIRON-TWO PIC X.

* THE FOLLOWING FIELDS ARE RECORD SELECTION ENTRIES
05 CFX-SELECT-COUNT PIC S9(3) COMP.
05 CFX-SELECT OCCURS 16 TIMES INDEXED BY CFX-S-INDX.
   10 CFX-SELECT-NAME PIC X(12).
   10 CFX-SELECT-OPER PIC X(2).
   10 CFX-SELECT-COMPARE PIC X(2).
   10 CFX-SELECT-VALUE PIC X(25).
   15 FILLER REDEFINES CFX-SELECT-VALUE.
   15 CFX-SELECT-VALNUM PIC S9(18) COMP-3.

* THE FOLLOWING FIELDS ARE EXIT PARAMETER ENTRIES
05 CFX-PARM-COUNT PIC S9(3) COMP.
05 CFX-PARMS OCCURS 16 TIMES INDEXED BY CFX-P-INDX.
   10 CFX-PARM-NAME PIC X(12).
   10 CFX-PARM-OPER PIC X(2).
   10 CFX-PARM-COMPARE PIC X(2).
   10 CFX-PARM-VALUE PIC X(25).
   10 FILLER REDEFINES CFX-PARM-VALUE.
   15 CFX-PARM-VALNUM PIC S9(18) COMP-3.
   15 FILLER PIC X(15).

* THE FOLLOWING FIELDS ARE EXIT PARAMETER ENTRIES
05 CFX-MESSAGE.
   10 CFX-ANSWER-TYPE PIC X.
   88 CFX-ANSWER-ANY VALUE 'A'.
   88 CFX-ANSWER-DARK VALUE 'O'.
   88 CFX-ANSWER-OLD VALUE 'N'.
   88 CFX-ANSWER-NEW VALUE 'O'.
   10 CFX-MESSAGES.
      15 CFX-MSG1 PIC X(76).
      15 FILLER PIC X(04).
      15 CFX-MSG2 PIC X(76).
      15 FILLER PIC X(04).
      15 CFX-MSG3 PIC X(76).
      15 FILLER PIC X(04).
      05 CFX-BUFFER-ITEM-SIZE PIC S9(4) COMP.
      05 CFX-BUFFER-ITEM-COUNT PIC S9(4) COMP.
      05 FILLER PIC X(12).
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFX-LANGUAGE</td>
<td>PIC X(1).</td>
</tr>
<tr>
<td>CFX-RESTART-PTR</td>
<td>PIC S9(7) COMP.</td>
</tr>
<tr>
<td>CFX-WP-TYPE-DATA</td>
<td>PIC X(01) VALUE SPACE.</td>
</tr>
<tr>
<td>CFX-ITS-WP-DATA</td>
<td>VALUE 'I'.</td>
</tr>
<tr>
<td>CFX-ITS-MIXED-CASE</td>
<td>VALUE 'M'.</td>
</tr>
<tr>
<td>CFX-TYPE</td>
<td>PIC X(01) VALUE SPACE.</td>
</tr>
<tr>
<td>CFX-ITS-BASIC</td>
<td>VALUE 'B'.</td>
</tr>
<tr>
<td>CFX-ITS-DIF</td>
<td>VALUE 'D'.</td>
</tr>
<tr>
<td>CFX-ITS-UPPER-CASE</td>
<td>VALUE 'U'.</td>
</tr>
<tr>
<td>CFX-ITS-TRANSPARENT</td>
<td>VALUE 'T'.</td>
</tr>
<tr>
<td>CFX-ITS-UPPER-CASE</td>
<td>VALUE 'U'.</td>
</tr>
<tr>
<td>CFX-WP-TYPE-DATA</td>
<td>VALUE 'I'.</td>
</tr>
<tr>
<td>CFX-ITS-WP-DATA</td>
<td>VALUE 'I'.</td>
</tr>
<tr>
<td>CFX-ITS-MIXED-CASE</td>
<td>VALUE 'M'.</td>
</tr>
<tr>
<td>CFX-ITS-BASIC</td>
<td>VALUE 'B'.</td>
</tr>
<tr>
<td>CFX-ITS-DIF</td>
<td>VALUE 'D'.</td>
</tr>
<tr>
<td>CFX-ITS-UPPER-CASE</td>
<td>VALUE 'U'.</td>
</tr>
<tr>
<td>CFX-ITS-TRANSPARENT</td>
<td>VALUE 'T'.</td>
</tr>
<tr>
<td>CFX-ITS-UPPER-CASE</td>
<td>VALUE 'U'.</td>
</tr>
<tr>
<td>CFX-WP-TYPE-DATA</td>
<td>VALUE 'I'.</td>
</tr>
<tr>
<td>CFX-ITS-WP-DATA</td>
<td>VALUE 'I'.</td>
</tr>
<tr>
<td>CFX-ITS-MIXED-CASE</td>
<td>VALUE 'M'.</td>
</tr>
<tr>
<td>CFX-ITS-BASIC</td>
<td>VALUE 'B'.</td>
</tr>
<tr>
<td>CFX-ITS-DIF</td>
<td>VALUE 'D'.</td>
</tr>
<tr>
<td>CFX-ITS-UPPER-CASE</td>
<td>VALUE 'U'.</td>
</tr>
<tr>
<td>CFX-ITS-TRANSPARENT</td>
<td>VALUE 'T'.</td>
</tr>
<tr>
<td>CFX-ITS-UPPER-CASE</td>
<td>VALUE 'U'.</td>
</tr>
</tbody>
</table>

**Figure C-1: COBOL Control Block**
Sample Exits - Data Base Access
The figures shown below are samples of using the user exit to access databases in DL/I and IDMS. The samples are skeleton programs to be used as guidelines. They are not complete programs. These are shown in Figures C-2 and C-3. Figures C-4 through C-8 show the profile required to use these exits.

*********** DL/I Sample Exit Program for pcMAINFRAME ***********

IDENTIFICATION DIVISION.
PROGRAM-ID. 'SLSEXTCT'.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
  77 PSB-NAME PICTURE X (8) VALUE 'SLSPSB'.
  77 FUNC-SCHED PICTURE XXXX VALUE 'PCB '.
  77 FUNC-TERM PICTURE XXXX VALUE 'TERM'.
  77 FUNC-GN PICTURE XXXX VALUE 'GN '.
  77 AGENT-SSA PICTURE X(11) VALUE 'BRANCH *D'.
  01 MESSAGES.
    05 WS-MSG1 PICTURE X(60) VALUE 'UNABLE TO ACCESS SALES MASTER. CONTACT HOST SITE'.
    05 WS-MSG2 PICTURE X(60) VALUE 'ABNORMAL DL/I RESPONSE. TRANSMISSION TERMINATED'.
    05 WS-MSG3 PICTURE X(60) VALUE 'INVALID PCMAINFRAME REQUEST. ABORTED'.
* NOTE: THE BRANCH AND AGENT SEGMENTS MUST BE CONTIGUOUS SINCE A PATH CALL IS USED TO RETRIEVE THEM *
  01 BRANCH-SEG.
    05 BR-BRANCH-NBR PICTURE X(5).
  01 AGENT-SEG.
    05 AG-AGENT-NBR PICTURE X(5).
  ** FORMAT: OYYYDDD (JULIAN)
    05 AG-YTD-SALES PICTURE S9(7) COMP-3.
  01 PCMF-RECORD.
    05 PCMF-BRANCH PICTURE X(5).
    05 PCMF-AGENT PICTURE X(5).
    05 PCMF-YTD-SALES PICTURE S9(7) COMP-3.

LINKAGE SECTION.
  01 DFHCOMMAREA.
COPY CFXCOMX.
  05 FILLER REDEFINES CFX-WORK.
  10 BLL-SAVE.
  20 SAVE-UIB-BLL PICTURE S9(8) COMP.
  20 SAVE-PCB-PTRS-BLL PICTURE S9(8) COMP.
* NOTE: CFX-WORK MUST BE THE REPOSITORY FOR ANY AND ALL VARIABLES WHICH MUST BE PASSED FROM ONE ENTRY TO THIS PROGRAM TO THE NEXT. WORKING-STORAGE MUST NOT BE USED FOR THIS PURPOSE, SINCE EACH ENTRY TO THIS PROGRAM WILL RECEIVE A FRESH COPY OF WORKING STORAGE. *
  20 SAVE-PCB1-BLL PICTURE S9(8) COMP.

  01 BLLCELLS.
    05 FILLER PICTURE S9(8) COMP.
    05 UIB-BLL PICTURE S9(8) COMP.
    05 PCB-PTRS-BLL PICTURE S9(8) COMP.
    05 PCB1-BLL PICTURE S9(8) COMP.
01 DLUIB.
COPY DLUIB.

01 PCB-PTRS.
   02 B-PCB1-PTR PICTURE S9(8) COMP.

01 PCB1.
   02 PCB1-DBD-NAME PICTURE X(8).
   02 PCB1-SEG-LEVEL PICTURE XX.
   02 PCB1-STATUS-CODE PICTURE XX.
   02 PCB1-PROC-OPT PICTURE XX.
   02 FILLER PICTURE S9(5) COMP.
   02 PCB1-SEG-NAME PICTURE X(8).
   02 PCB1-LEN-KFB PICTURE S9(5) COMP.
   02 PCB1-MU-SENSEG PICTURE S9(5) COMP.
   02 PCB1-KAY-FB PICTURE X(256).

PROCEDURE DIVISION.

PROGRAM-ENTRY.
   IF CFX-OPEN
      PERFORM PROCESS-OPEN-REQUEST
      THRU PROCESS-OPEN-REQUEST-EXIT
   ELSE
      IF CFX-READ
         PERFORM PROCESS-READ-REQUEST
      THRU PROCESS-READ-REQUEST-EXIT
   ELSE
      IF CFX-CLOSE
         PERFORM PROCESS-CLOSE-REQUEST
      THRU PROCESS-CLOSE-REQUEST-EXIT
   ELSE
      MOVE 'F' TO CFX-RETURN-CODE
      MOVE WS-MSG1 TO CFX-MSG1.
   EXEC CICS RETURN END-EXEC.

PROCESS-OPEN-REQUEST.
   CALL 'CBLTDLI' USING FUNC-SCHED, PSB-NAME, UIB-BLL.
   * IF DL/I SCHEDULING FAILED, PASS MESSAGE BACK A MESSAGE
   * AND A BREAK-REQUEST RETURN CODE TO PC-MAINFRAME.
   IF UIBFCTR NOT = LOW-VALUES
      MOVE 'B' TO CFX-RETURN-CODE
      MOVE WS-MSG1 TO CFX-MSG1
      GO TO PROCESS-OPEN-REQUEST-EXIT.
   * SCHEDULING SUCCESSFUL. ESTABLISH POINTERS.
   MOVE UIBPCBAL TO PCB-PTRS-BLL.
   MOVE B-PCB1-PTR TO PCB1-BLL.
   * POINTERS MUST NOW BE SAVED FOR SUBSEQUENT USE
   * WHEN PCMAINFRAME NEXT ENTERS THIS PROGRAM.
   MOVE UIB-BLL TO SAVE-UIB-BLL.
   MOVE PCB-PTRS-BLL TO SAVE-PCB-PTRS-BLL.
   MOVE PCB1-BLL TO SAVE-PCB1-BLL.
   MOVE '0' TO CFX-RETURN-CODE.

PROCESS-OPEN-REQUEST-EXIT.
EXIT.

PROCESS-READ-REQUEST.
   MOVE SAVE-UIB-BLL TO UIB-BLL.
   MOVE SAVE-PCB-PTRS-BLL TO PCB-PTRS-BLL.
   MOVE SAVE-PCB1-BLL TO PCB1-BLL.
   PERFORM GET-NEXT-AGENT.
   * IF UIBFCTR NOT = LOW-VALUES
   OR (PCB1-STATUS-CODE NOT = 'GB' AND NOT = ' ')
   * ABNORMAL DL/I STATUS RETURNED
      MOVE 'F' TO CFX-RETURN-CODE
      MOVE WS-MSG2 TO CFX-MSG1
      GO TO PROCESS-READ-REQUEST-EXIT.
IF PCB1-STATUS-CODE = 'GB'
   * END OF DATABASE
   MOVE 'E' TO CFX-RETURN-CODE
   CALL 'CBLTDLI' USING FUNC-TERM
   GO TO PROCESS-READ-REQUEST-EXIT.
   MOVE 'O' TO CFX-RETURN-CODE.
   MOVE BR-BRANCH-NBR TO PCMF-BRANCH.
   MOVE AG-AGENT-NBR TO PCMF-AGENT.
   MOVE AG-YTD-SALES TO PCMF-YTD-SALES.
   MOVE PCMF-RECORD TO CFX-RECORD.
   PROCESS-READ-REQUEST-EXIT.
   EXIT.
   GET-NEXT-AGENT.
   CALL 'CBLTDLI' USING FUNC-GN, PCB1,
      AGENT-SEG, AGENT-SSA.
   IF UIBFCTS = LOW-VALUES
      IF PCB1-STATUS-CODE = SPACES
         AND AG-START-DATE < CFX-PARM-VALNUM (1)
         GO TO GET-NEXT-AGENT.

Figure C-2: DL/I Sample Exit
*************** IDMS SAMPLE EXIT FOR pcMAINFRAME ***************

IDENTIFICATION DIVISION.
PROGRAM-ID. 'SLSEXTCT'.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.

******************************************************************************
* NOTE: IDMS WORKING STORAGE DEFINITIONS OTHER THAN RECORD *
* DEFINITIONS VARY FROM INSTALLATION TO INSTALLATION AND *
* ARE NOT INCLUDED HERE. CODE THEM AS YOU WOULD FOR ANY *
* OTHER CICS COMMAND LEVEL IDMS APPLICATION. *
******************************************************************************

01 MESSAGES.
  05 WS-MSG1  PICTURE X(60) VALUE 'UNABLE TO ACCESS SALES MASTER. CONTACT HOST SITE'.
  05 WS-MSG2  PICTURE X(60) VALUE 'ABNORMAL IDMS RESPONSE. TRANSMISSION TERMINATED'.
  05 WS-MSG3  PICTURE X(60) VALUE 'INVALID PCMAINFRAME REQUEST. ABORTED'.

01 BRANCH.
  05 BR-BRANCH-NBR  PICTURE X(5).

01 AGENT.
  05 AG-AGENT-NBR  PICTURE X(5).
  05 AG-START-DATE  PICTURE S9(7) COMP-3.  * FORMAT: OYYYDDD (JULIAN)
  05 AG-YTD-SALES   PICTURE S9(7) COMP-3.

01 PCMF-RECORD.
  05 PCMF-BRANCH  PICTURE X(5).
  05 PCMF-AGENT   PICTURE X(5).
  05 PCMF-YTD-SALES PICTURE S9(7) COMP-3.

LINKAGE SECTION.
01 DFHCOMMAREA.
COPY CFXCOMX.
  05 FILLER REDEFINES CFX-WORK.
   10 NEW-BRANCH-REQUIRED-SW  PICTURE X.
   88 NEW-BRANCH-REQUIRED VALUE 'Y'.
   10 FIRST-IDMS-CALL-SW     PICTURE X.
   88 FIRST-IDMS-CALL VALUE 'Y'.
   10 SAVE-BRANCH           PICTURE X(5).

******************************************************************************
* NOTE: CFX-WORK MUST BE THE REPOSITORY FOR ALL VARIABLES WHICH *
* MUST BE PASSED FROM ONE ENTRY TO THIS PROGRAM TO THE NEXT. *
******************************************************************************

PROCEDURE DIVISION.
PROGRAM-ENTRY.

   IF CFX-OPEN
      PERFORM PROCESS-OPEN-REQUEST
      THRU PROCESS-OPEN-REQUEST-EXIT
   ELSE
   IF CFX-READ
      PERFORM PROCESS-READ-REQUEST
      THRU PROCESS-READ-REQUEST-EXIT
   ELSE
   IF CFX-CLOSE
      PERFORM PROCESS-CLOSE-REQUEST
      THRU PROCESS-CLOSE-REQUEST-EXIT
   ELSE
      MOVE 'F' TO CFX-RETURN-CODE
      MOVE WS-MSG3 TO CFX-MSG1.
   END-IF
   EXEC CICS RETURN-END-EXEC.
PROCESS-OPEN-REQUEST.
BIND RUN-UNIT.

*************************************************
* NOTE: RECORD "BINDS" MAY NOT BE DONE HERE, BECAUSE A NEW
* COPY OF WORKING STORAGE IS PROVIDED BY CICS EACH TIME THE
* PROGRAM IS ENTERED. INSTEAD, THEY ARE DONE AT READ TIME.
*************************************************

IF IDMS-STATUS NOT = '0000'
MOVE 'B' TO CFX-RETURN-CODE
MOVE WS-MSG1 TO CFX-MSG1
GO TO PROCESS-OPEN-REQUEST-EXIT.

READY SALES-AREA.

IF IDMS-STATUS NOT = '0000'
MOVE 'B' TO CFX-RETURN-CODE
MOVE WS-MSG1 TO CFX-MSG1
GO TO PROCESS-OPEN-REQUEST-EXIT.

MOVE 'O' TO CFX-RETURN-CODE.
MOVE 'Y' TO NEW-BRANCH-REQUIRED-SW, FIRST-IDMS-CALL-SW.
GO TO PROCESS-OPEN-REQUEST-EXIT-EXIT.

PROCESS-READ-REQUEST.
BIND BRANCH.
BIND AGENT.

IF NEW-BRANCH-REQUIRED
MOVE 'N' TO NEW-BRANCH-REQUIRED-SW
PERFORM OBTAIN-NEXT-BRANCH
IF IDMS-STATUS = '0000'
MOVE ER-BRANCH-NBR TO SAVE-BRANCH
ELSE
IF IDMS-STATUS = '0307'
END OF DATABASE
FINISH
MOVE 'E' TO CFX-RETURN-CODE
GO TO PROCESS-READ-REQUEST-EXIT
ELSE
ABNORMAL IDMS STATUS RETURNED
MOVE 'F' TO CFX-RETURN-CODE
MOVE WS-MSG2 TO CFX-MSG1
GO TO PROCESS-READ-REQUEST-EXIT.

PERFORM OBTAIN-NEXT-AGENT.

IF (IDMS-STATUS NOT = '0000' AND NOT = '0307')
ABNORMAL IDMS STATUS RETURNED
MOVE 'F' TO CFX-RETURN-CODE
MOVE WS-MSG2 TO CFX-MSG1
GO TO PROCESS-READ-REQUEST-EXIT.

MOVE 'O' TO CFX-RETURN-CODE.
MOVE SAVE-BRANCH TO PCMF-BRANCH.
MOVE AG-AGENT-NBR TO PCMF-AGENT.
MOVE AG-YTD-SALES TO PCMF-YTD-SALES.
MOVE PCMF-RECORD TO CFX-RECORD.
GO TO PROCESS-READ-REQUEST-EXIT.

OBTAIN-NEXT-BRANCH.
IF FIRST-IDMS-CALL
OBTAIN FIRST BRANCH WITHIN SEQ-BRANCH
MOVE 'N' TO FIRST-IDMS-CALL-SW
ELSE
OBTAIN NEXT BRANCH WITHIN SEQ-BRANCH.

OBTAIN-NEXT-AGENT.
OBTAIN NEXT AGENT WITHIN BRANCH-AGENT.

IF IDMS-STATUS = '0000'
AND AG-START-DATE < CFX-PARM-VALNUM (1)
GO TO OBTAIN-NEXT-AGENT
ELSE
IF IDMS-STATUS = '0307'
MOVE 'Y' TO NEW-BRANCH-REQUIRED-SW.

Figure C-3: IDMS Sample Exit
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Maximum</th>
<th>Data</th>
<th>Dec</th>
<th>Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTDATE</td>
<td></td>
<td>N</td>
<td></td>
<td>&amp;STARTDATE</td>
</tr>
</tbody>
</table>

Profile found - Update assumed.

Figure C-4: Exit Profile Example

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Maximum</th>
<th>Data</th>
<th>Dec</th>
<th>Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTDATE</td>
<td></td>
<td>N</td>
<td></td>
<td>&amp;STARTDATE</td>
</tr>
</tbody>
</table>

Profile found - Update assumed.

Figure C-5: Exit Profile Example
### Appendix C - Coding User Exits System Administrator's Guide

#### Profile: DBEXIT
Profile Definition

<table>
<thead>
<tr>
<th>Data/Field</th>
<th>I Begin</th>
<th>Data Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>D Pos</td>
<td>Bytes</td>
</tr>
</tbody>
</table>

Define Include/Exclude rules...
from none to 12 are allowed.

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete.
PF1 - 1st page, PF2 - Next Page, PF3 - Prior Page, CLEAR - Menu.

**Figure C-6: Exit Profile Example**

---

#### Profile: DBEXIT
Profile Definition

<table>
<thead>
<tr>
<th>Data/Field Begin</th>
<th>Data Dec Break</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Pos</td>
</tr>
</tbody>
</table>

Define rules for the break - If any rule is true the output break will occur.
From none to six rules are allowed.

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete.
PF1 - 1st page, PF2 - Next Page, PF3 - Prior Page, CLEAR - Menu.

**Figure C-7: Exit Profile Example**

---

#### Profile: SAMPLE
Profile Definition

<table>
<thead>
<tr>
<th>Data/Field Begin</th>
<th><strong>MF</strong></th>
<th><strong>PC</strong></th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Pos</td>
<td>Size</td>
<td>Type</td>
</tr>
<tr>
<td>BRANCH</td>
<td>1</td>
<td>5</td>
<td>C</td>
</tr>
<tr>
<td>AGENT</td>
<td>6</td>
<td>5</td>
<td>C</td>
</tr>
<tr>
<td>YTDSALES</td>
<td>11</td>
<td>4</td>
<td>P</td>
</tr>
</tbody>
</table>

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete.
PF1 - 1st page, PF2 - Next Page, PF3 - Prior Page, CLEAR - Menu.

**Figure C-8: Exit Profile Example**
Appendix D - VSE Power Access

CFXPOWEX is a user exit coded by cfSOFTWARE that gives you the ability to:

1. Upload/download individual POWER queue entries;
2. Upload/download a generic group of queue entries;
3. Download a queue listing for a particular POWER jobname;
4. Issue a POWER display command (PDISPLAY), with the resulting listing downloaded to a PC file.

To use CFXPOWEX, your mainframe must be running DOS VSE SP version 2.1 or above, and VSE POWER version 2.2 or above.

As with other types of file transfers, you must describe your requirements in a pcMAINFRAME profile. The profile parameters are the same as for any other type of transfer, but you must also:

1. Enter CFXPOWEX as the Exit Name and EXIT as the File Type on the first Profile Definition screen;
2. Supply the required parameters on the "User Exit Parameters" Profile Definition Screen. These parameters define the POWER job name, class, and other related information required to accomplish the transfer.

For VSE/POWER transfers, the maximum record size allowable is 1024 characters. If larger records occur in the data, they are truncated.

CFXPOWEX supports Fixed, Spool, Wks, xBASE, Excel or Transparent data transfers, however transparent format is supported only for PUN.

"Spool" format is used with the CFXPOWEX exit for transfers to and from the POWER LST queue. When Spool is used, word processing mode is assumed. Upper and lower case characters are transmitted intact — no upper case translation occurs. However, if you want uploaded Spool data translated to upper case (often a requirement for printing on mainframe printers), use the exit parm UCTRAN (see the following section). In addition, the report's page breaks and spacing are maintained on the target system. During downloads, ASCII form feeds are inserted where needed. During uploads, skip-to-channel-one machine commands are used.
## CFXPOWEX Exit Parms for Queue Transfers

**ACCTYPE** *Optional.* Specify RESTRICTED for restricted access, UNRESTRICTED for unrestricted access.

RESTRICTED means the user is capable of downloading only POWER queue entries that have a TOUSER specification which matches the supplied USERID parameter. If no USERID parameter is supplied, only queue entries with TOUSER "ANY" may be downloaded.

UNRESTRICTED means the user may download any POWER queue entry, regardless of its destination and also without regard to password. Choose this option only after careful consideration.

If this parameter is not supplied, RESTRICTED is assumed. (alphanumeric, 10-12 characters)

**CLASS** *Required* for all but upload to Reader. Specify the VSE POWER class for the queue entry (alphanumeric, one character).

**COPY** *Optional.* Used for upload to List and Punch only. Specify the number of copies of output that will be printed or punched (numeric, 1-3 characters, 1-255).

**DESTNODE** *Optional.* Used for upload to List or Punch only. This is used to route the list or punch output to another node in the system (alphanumeric, 1-8 characters).

**DESTUSER** *Optional.* Used for upload to List or Punch only. This is used to route the list or punch output to another user in the system (alphanumeric, 1-8 characters).

**DISP** *Optional* for all uploads; not used for download. Specify the disposition to be used for the uploaded entry ("D", "K", "L", "H"). If not specified, "D" will be used.

**EXTFORCE** *Optional.* Used for downloading only. This forces pcMAINFRAME to build an extension to the filename for the file(s) downloaded.

If Parm Value = 'Y', the extension will comprise of a two digit number that is the low order two digits of the POWER job number and an optional third character that is the segment number if the job is segmented.

If Parm Value = '=XXX', the extension will be developed as:

- XXX - for the first occurrence of a given job name
- XX1 - XX9 - for duplicate jobs 2 through 10
- X10 - X99 - for duplicate jobs 11 through 100
- 100 - 998 - for duplicate jobs 101 through 999

**FCB** *Optional.* Used for upload to List or Punch only. This specifies the forms control buffer (FCB) to be used printing/punching the entry that has been uploaded (alphanumeric, 1-8 characters).

**FNO** *Optional.* Used for upload to Punch or List only. Specify the form ID to be used for the uploaded entry (alphanumeric, 1-4 characters).

**INITIALFF** *Optional.* Used for uploading only. If specified, a form feed is inserted into the beginning of the file prior to uploading. This may be
used to keep a PC generated report from printing on the same page as the job separator or JCL when it is uploaded.

**JOBNAME Required** for all transfers except upload to Reader. Specify jobname for the queue entry (alphanumeric, 1-8 characters). For generic transfers (that is, transfers where a "wildcard" PC filename specification has been used), JOBNAME must be omitted. For downloads, all entries to be downloaded should have POWER disposition "D" or "K".

**JOBNUMBER Optional.** Used for download only. Specify the POWER-assigned job number for the entry to be downloaded (numeric, 1-5 characters). This parameter is required if you have more than one job in the POWER queue with the same name and class, and you wish to download other than the first such queue entry. For generic transfers, JOBNUMBER must be omitted.

**JSEP Optional.** Used for upload to List or Punch only. This specifies the number of separator pages/cards to be used in the output (numeric, 1-2 characters). If this parameter is zero or omitted, the POWER-defined system default will be used.

**PAD Optional.** Used for download from the List queue only. Report records will be padded with spaces after extract from the queue and before record select and extract rules are applied. The default PAD length is 132. If larger records are needed, use the PAD parameter. (numeric, 4 characters).

**PASSWORD Optional.** Specify the password to be associated with the uploaded entry, or the required password for the entry to be downloaded (alphanumeric, 1-8 characters).

**PCCTL Optional** (alphanumeric, 3 characters). For uploads, specify "ASA" if the first character of the records in your input PC file contains an ASA control character that you wish to be honored in the output spool entry.

For downloads, the records of your downloaded PC file will contain an ASA control character in the first record position if **PCCTL=ASA**.

**PCNAME Optional** (alphanumeric, 6 characters). For downloads only, allows for specification of how the pcfilename name will be developed.

- **TOUSER** indicates that the destination user should be used as the PCFILE name.
- **JOBSEQ** indicates that the last three bytes of the job name should be used to carry a sequence number in case of duplicate job names. This parameter is commonly used with EXTFORCE.

**PRI Optional** for all uploads; not used for download. Specify the priority to be used for the uploaded entry (numeric, 1 character). If omitted, the POWER-defined system default will be used.

**PURGE Optional.** Used for download only from List/Punch. Specify "Y" or "N" ("N" is the default).

If PURGE=Y, queue entries with POWER disposition "D" will be deleted after download; queue entries with POWER disposition "K" will revert to disposition "L".

Caution must be used when specifying PURGE=Y with queue entries that have DISPOSITION "D"; use it only if the job output can be easily
re-created. Particular caution must be used if ACCTYPE is UNRESTRICTED.

**QUEUE Required.** Specify R, L, or P for Reader, List or Punch, respectively. If R is specified for an upload, your source file must be a valid VSE/POWER JECL jobstream. Your installation may employ a VSE RDREXIT to enforce standards for JECL, and the RDREXIT may "flush" jobs which do not meet standards. You will receive no pcMAINFRAME message if this happens, since the standard IBM VSE/POWER interface employed by pcMAINFRAME does not report this event.

**QUEUELST Optional.** Used for download only. Specify "Y" to download a list of jobs in the POWER queue which match your supplied JOBNAME, CLASS, and optionally, JOBNUMBER. The data in the entry is not downloaded.

Specify "N", or omit this parameter, if you wish to download the actual data in the queue.

Obtaining a list in this fashion is useful if you don’t know if the output you want is available, or if you have more than one entry in the queue and you need to know the associated POWER-assigned job numbers.

**SUFFIX Optional.** Used for download only. Specify the output suffix (segment) number for the entry to be downloaded. Segments are numbered from 1 to "n". If not specified, the first output segment is assumed (numeric, 1-3 characters). For generic transfers, SUFFIX must be omitted.

**SYSID Optional.** Used for upload only. Specify the target system on which the queue entry is to be processed, printed, or punched (alphanumeric, 1 character).

**UCTRAN Optional.** Used for upload only. Specify "Y" (the default), if you want lower-case characters to be translated to upper case. This is useful if your mainframe printer is unable to print lower-case data.

**USERID Optional.** Specify the USERID associated with the entry to be uploaded or downloaded (alphanumeric, 1-8 characters).

This is the user-id sub-parameter of the DEST parm in * $$ LST statement, or the LDEST parameter in the * $$ JOB statement. In the following two examples, USR1 is the USERID.

* $$ JOB JNM=JOB1,CLASS=0,LDEST=(,USR1)

* $$ LST CLASS=A,DISP=K,DEST=(*,USR1)

For downloads, USERID is required unless ACCTYPE is UNRESTRICTED or the job’s output has been directed to USERID "ANY".

For uploads, this specifies the "from" user, not the target user ID (see DESTUSER). Thus, for uploads USERID is of limited importance and may normally be omitted.

**Note:** Do not specify DESTNODE or DESTUSER if your uploaded data is to be printed or punched by your local mainframe computer.

The remaining user exit parms all specify information that will be printed on the job’s VSE/POWER separator pages/cards; they are only useful for uploads. Even though
all items are optional with regard to pcMAINFRAME, your installation may have
standards that require you to specify one or more of these:

- **USERINFO** Alphanumeric, 1-16 characters
- **BUILDING** Alphanumeric, 1-8 characters
- **DEPARTMENT** Alphanumeric, 1-8 characters
- **ROOM** Alphanumeric, 1-8 characters
- **PROGRAMMER** Alphanumeric, 1-20 characters

When uploading files to the LST queue for printing on an IBM 3800 printer, the
following additional parameters are supported:

- **BURST** Y|N
- **CHARS** tablename,[tablename...]
- **COPYG** groupvalue[,groupvalue...]
- **DFLT** Y|N
- **FLASH** [overlayname][.count]
- **MODIFY** copymodname[,tablename]

All the 3800 parameters are alphanumeric. Note that their syntax is identical to the
corresponding keyword values used in the VSE/POWER LST JECL statement, with
the exception that parentheses are not used to enclose the **CHARS**, **COPYG**, **FLASH**, or **MODIFY** parameters. Consult your VSE/POWER documentation for
details.

**CFXPOWEX Exit Parms for Command Processing**

**PWRCMD** When specified, this exit parameter must be the only exit parameter
supplied. It specifies the command to be executed (which must be a display
command). Examples:

```
D LST,*ACCTG
PDISPLAY PUN,*COMPILE
D PUN,FULL=YES
```

The results are downloaded to the PC.

**Generic Transfers**

Generic transfers are transfers where you have specified "wildcard" characters (*, ?)
in the PC filename. Generic uploads typically cause more than one PC file to be
uploaded to the POWER queue; generic downloads usually result in more than one
PC file being created.

For downloads, all queue entries in the specified queue (RDR, PUN, LST) are
examined. An individual queue entry is eligible for download only if:

1. its eight-character jobname matches the wildcard PC file specification;
2. its Disposition is "D" or "K";
3. its Class matches the exit profile’s CLASS specification.

If ACCTYPE is RESTRICTED (or defaulted to RESTRICTED), the queue entries
must also match the provided USERID exit parameter, and, if supplied, the
PASSWORD parameter. PURGE=Y may be included in your exit parameters, in which case it is applied to each downloaded entry.

In cases where multiple POWER queue entries which have the same POWER jobname are downloaded, pcMAINFRAME uses the low-order two characters of the POWER job number as the PC file extension for all but the first such file. In this way, multiple POWER entries with the same name can be downloaded to the same PC directory without conflict.

A maximum of 35 segments are downloaded per job. For downloaded multi-segment output, a third character is added to the PC file extension, and identifies the segment number (1-9 for segments 1-9, A-Z for segments 10-35).

A maximum of 150 queue entries can be downloaded in a single generic request.

Sample Profiles

Included with your pcMAINFRAME installation material are ten sample profiles which illustrate the most common types of spool access:

- **POWER01** - Download a report from VSE/POWER
- **POWER02** - Upload a report to VSE/POWER
- **POWER03** - Download punch from VSE/POWER
- **POWER04** - Upload punch to VSE/POWER
- **POWER05** - Upload jobstream to VSE/POWER RDR
- **POWER06** - Upload binary file to VSE/POWER PUN
- **POWER07** - Download binary file from POWER PUN
- **POWER08** - Download a LST QUEUELIST
- **POWER09** - Issue a PDISPLAY command, download the results
- **POWER10** - Download generic from punch
- **POWER11** - Download a report from VSE/POWER with ASA PC carriage control

Upload/Download Specifications

Upload

For all data types, the machine command code employed is X'01' for the punch, and X'09' (write before spacing one) for list.

Also for all data types, the maximum record length for upload/download with is 1024. If a larger record occurs in the data, it is truncated to the maximum size and a warning message is issued at the PC (processing continues).

Spool Data:

- Word processing mode is assumed
- Form feeds embedded in PC files are honored and converted to Skip-to-CH01 machine commands
- If UCTRAN=Y (exit parm) has been specified, lower-case characters are converted to upper case.
- Non-printable characters are converted to spaces.
- Trailing blanks are removed from records before they are written
Transparent Data:

- Supported only for PUN queue.
- No mainframe channel control commands are used (no top-of-page commands are stored in the queue entry).
- All characters are stored "as-is" per the normal convention for transparent data. No translation takes place.
- Trailing blanks are not removed from records before they are stored; they are stored exactly as-is.
- LRECL must be set to 80.

Fixed, BASIC, DIF:

- No mainframe channel control commands are used (listings will be single-spaced).
- If UCTRAN=Y and word processing is specified, lower-case characters are converted to upper case by the power exit; pcMAINFRAME will convert to upper-case regardless of UCTRAN if word processing is not specified.
- If word processing is not specified, non-printable characters are converted to blanks.
- If word processing is specified, non-printable characters are stored as-is.
- Trailing blanks are removed from records before they are written.

Download

Spool Data:

- Word processing mode is assumed.
- Skip-to-CH01 commands, SPACE 1, 2, or 3 before and immediate commands are honored for download from LST. The downloaded PC file will use CR, LF, and FF control characters as necessary to simulate the mainframe report format. Immediate commands other than these (for example: Skip to channel 2 immediate) are ignored.
- For download from PUN and RDR, channel and spacing commands are ignored (single-spacing is used for all text lines).
- A CTRL-Z character will be appended to the downloaded file.
- UCTRAN=Y (exit parm) has no effect.
- Non-printable characters are converted to spaces.
- Trailing blanks and nulls are removed from records before they are written; if the line is all nulls or all blank, however, a single space will be output (this avoids passing a zero-length record to pcMAINFRAME).

Transparent Data:

- Supported only for PUN queue.
- Mainframe channel and spacing commands are ignored.
- Non-printable characters are stored "as-is" per the normal convention for transparent data.
- Trailing blanks are not removed from records before they are stored; they are stored exactly as-is.
An LRECL of 80 must be specified.

**Fixed, BASIC, DIF:**

- Mainframe channel and spacing commands are ignored; all text lines will be single-spaced; pcMAINFRAME, not the POWER exit, will store CR/LF characters as appropriate (as well as pad/truncate short and long records if a fixed LRECL is defined in the profile).
- UCTRAN=Y (exit parm) has no effect.
- If word processing is not specified, non-printable characters are converted to blanks.
- If word processing is specified, non-printable characters are stored as-is.
- Records downloaded from PUN/RDR are padded to 80 characters; records downloaded from LST are padded to 132 characters.

**Useful VSE/POWER Commands**

If pcMAINFRAME abnormally terminates for any reason while in the process of accessing spool data, a POWER SAS (spool-access support) task may be left active.

If you suspect this is the case, your mainframe computer operator can enter the following VSE/POWER command:

```
D A,SAS  (display active SAS tasks)
```

VSE/POWER will respond with messages similar to the following:

```
1R48I  SAS,taskno,SAS=CFXPOWRc,R000,REQ=PUT, jobname, jobnumber rrrrr RECORDS SPOOLED
```

where:

- `taskno` - is the SAS task number
- `jobname` - is the name of the POWER job name
- `rrrrr` - is the number of records spooled
- `jobnumber` - is the POWER job number

This task may be terminated by the mainframe computer operator by issuing the following command:

```
P SAS,taskno
```

where:

- `taskno` - is the SAS task number

VSE/POWER should respond with the following message:

```
1QX3I CROSS-PARTITION TASK taskno SERVING SAS=CFXPOWRc STOPPED
```

It is imperative that you verify that this is not a normally-running pcMAINFRAME upload/download before the task is terminated in this fashion.
Appendix E - JES Spool Access

CFXJESEX is a user exit supplied by cfSOFTWARE that allows you to transfer data between your PC and your mainframe’s JES spool queues.

To use CFXJESEX, your mainframe must be running CICS version 1.7 or above. In addition, your CICS system must be defined to support the standard CICS interface to JES (DFHSIT SPOOL=YES must be specified).

As for other types of file transfer, you must describe your requirements in a pcMAINFRAME profile. The profile parameters are the same as for any other type of transfer, but you must also:

1. Enter CFXJESEX as the Exit Name and EXIT as the File Type on the first Profile Definition screen;

2. Supply the required parameters on the User Exit Parameters - Profile Definition Screen. These parameters define the external writer name and other related information required to accomplish the transfer (see "CFXJESEX Exit Parms" below).

For spool access, the maximum record size allowable is 132 characters (133 if PCCTL is ASA). If larger records occur, they are truncated.

All pcMAINFRAME data formats are supported for use with the CFXJESEX. For all but very unusual situations, however, one of three types — Fixed, Spool, or Transparent — will be appropriate. Fixed and Transparent data types have the same meaning and implications as they do for other pcMAINFRAME processing. It is possible to upload transparent binary files to be downloaded by other users in your installation.

"Spool" format has been provided for exclusive use with spool access. It is intended primarily for use when uploading or downloading reports. When Spool is used, word processing mode is assumed — upper and lower case characters are transmitted intact. No upper case translation occurs. If you wish for uploaded Spool data to be translated to upper case (often a requirement for printing on mainframe printers), use the exit parm UCTRAN. In addition, the report’s page breaks and spacing are maintained on the target system. For downloads, ASCII form feeds are inserted where needed; for uploads, skip-to-channel-one printer commands are used.
**Downloading Reports**

In order to be downloaded, a mainframe report must be:

1. Released and ready for printing (the output may not be "held");
2. Associated with a JES external writer name, the first four characters of which are the same as the CICS Applid. This external writer name must be specified in the USERID exit parm (see "CFXJESEX Exit Parms").

The menu screen from the pcMAINFRAME administrative transaction displays the CICS Applid, the first four characters of this Applid must match the first four characters of the USERID parameter and the JES external writer name.

Normally, the external writer name is specified in the OS run JCL for the job which creates the output, for example:

```plaintext
//STEP5.SYSPRINT DD SYSOUT=(X,CICSCFX)
```

This statement would direct SYSOUT in STEP5 to class X, external writer CICSCFX; for downloading, CICSCFX would also be specified in the USERID exit parm within the pcMAINFRAME profile.

We strongly recommend that you standardize these writer names, and that you use a unique writer name for each target PC user. One approach would be to use the pcMAINFRAME PC ID or the CICS terminal ID as the last four characters of the writer name.

Proper formatting of the report on the PC requires you to correctly specify the type of printer control character present in the output (normally ASA); this is done via the pcMAINFRAME profile user exit parameters which are described later in this section.

We recommend that you use a different printer output class for each printer control character type (ASA, Machine, None), and that you specify the CLASS exit parameter to restrict download for the profile to the desired class. This may enable you to use the same profile for all reports having that control character type. It should also prevent you from accidentally downloading a report whose printer control character type differs from that specified in the profile.

**Reducing Download Bottlenecks with TSLIMIT**

CFXJESEX uses standard IBM-supplied interfaces when accessing the JES queues. The provided interfaces allow only one user at a time to read the JES queues, though they permit multiple users to concurrently write to the queues.

If the download interface is busy when a download is attempted, CFXJESEX periodically tries again to gain control of the interface (see RETRY and RETRYINT in "CFXJESEX Exit Parms", below). If large numbers of users are vying for reports, this remedy alone may be inadequate; in that case, consider adding TSLIMIT to your profiles.

If the TSLIMIT exit parameter is specified, each JES report is written to CICS temporary storage before it is downloaded. When the temporary storage queue is complete, the JES interface is freed for other users. Since writing the report to temporary storage is ordinarily much quicker than downloading the report directly, the interface is freed sooner than it would otherwise be.
In many shops, temporary storage is a limited resource; specify **TSLIMIT** only if JES download bottlenecks occur, and only if you have evaluated the additional temporary storage burden that will be imposed by this approach. Consult your CICS systems programmer for more information.

**Uploading Reports**

Uploaded reports carry the characteristics of the CICS region in which the upload operation occurs. These characteristics include: job name, job number, form ID, and others. The only attribute which you may alter is the job **CLASS** (see CLASS user exit parm).

If you are sending the report to another CICS user, this should pose no problem. If you intend to print the reports at the mainframe, we recommend that you:

1. Arrange with your computer operations group to use unique job classes for your uploaded reports.
2. Use the **BANNER**, **RTINFO1**, and **RTINFO2** to produce a banner page for each report to assist in its routing.

**Uploading to the Reader**

You may use this facility to upload JCL and data to the internal JES reader. Specify the parameters **DESTUSER** and **DESTNODE** in the USER EXIT parameter screen.

The **DESTUSER** parameter must be given a value of "INTRDR" and the **DESTNODE** parameter must have a value of either the node name of the local system or "*". The "*" may be used only with CICS PUTLEVEL 9004 or above.

The **//JOB** card in any submitted JCL should contain the **USER=** parameter in order to pick up the correct security. If the **USER=** parameter is left off, the security assigned to the CICS region will be the default.

**CFXJESEX ExitParms**

**Note:** Underlined values are the defaults.

- **USERID** Mandatory. Used for download only. Specify the USERID (external writer name) associated with the JES queue entry you wish to download (alphanumeric, 1-8 characters).
- **CLASS** Optional. Specify the job class for the spool entry (alphanumeric, one character). For download, omission of this parameter means that all output will be downloaded. Do not used "*" for class designation.
- **CCTYPE** Optional. Used for both upload and download. Specify the type of carriage control character associated with data records. Must be "A", "M", or "N" (for ANSI standard, machine, or none).

  For uploads, specifying "A" or "M" does not mean your PC file is expected to contain an ANSI or machine control character in the first data position; it simply indicates to CFXJESEX what type of control characters to use in the created JES dataset. For pcMAINFRAME release 4.1 and above, your input PC file may contain ASA control characters (see **PCCTL** parameter)

  For downloads with **SPOOL** format, this parameter should be specified as "A" or "M"; pcMAINFRAME release 4.1 and above will
automatically determine whether your specification is correct and change it if necessary (pcMAINFRAME will not take this action if you specify CCTYPE "N", however).

**PCCTL Optional** (alphanumeric, 3 characters). For uploads, specify "ASA" if the first character of the records in your input PC file contains an ASA control character that you wish to be honored in the output spool entry.

For downloads, the records of your downloaded PC file will contain an ASA control character in the first record position if PCCTL=ASA.

**ALLFILES Optional**. Used for download only. Specify "Y"(es) or "N"(o).

If ALLFILES=N, only one JES dataset will be downloaded. If ALLFILES=Y, all JES data sets for the specified USERID will be downloaded as a single PC file.

To use ALLFILES=Y, you must also specify PURGE=Y (see below).

All data sets to be downloaded should employ the same type of carriage control.

**PURGE Optional**. Used for download only. Specify "Y" or "N".

If PURGE=Y, the spool queue entry will be deleted after download. Extreme caution must be used when specifying PURGE=Y; use it only if the job output can be easily re-created.

**UCTRAN Optional**. Used for upload only. Specify "Y" if you want lower-case characters to be translated to upper case. This is useful if your mainframe printer is unable to print lower-case data.

**DESTNODE Optional**. Used for upload only. This is used to route the spool entry to another node in the system (alphanumeric, 1-8 characters). DESTNODE should be omitted if you are uploading a report to be printed by the mainframe computer.

If you are routing a report or file to another pcMAINFRAME user, this parameter must be the JES node for the target user’s CICS system (consult your mainframe systems programmer for this information).

**DESTUSER Optional**. Used for upload only. This is used to route the spool entry to another user in the system (alphanumeric, 1-8 characters). DESTUSER should be omitted if you are uploading a report to be printed by the mainframe computer.

If you are routing a report or file to another pcMAINFRAME user, this parameter must be identical to the target user’s USERID value.

**BANNER Optional**. Used for upload only. Specify "Y"(es) or "N"(o).

Specifies whether or not CFXJESEX should produce banner pages for the uploaded report. This parameter will be ignored unless SPOOL is specified in the pcMAINFRAME profile.

**RTINFO1 Optional**. Used for upload only. Specify any information which will assist in routing the printed report. This information will be printed on the pcMAINFRAME banner page for the report.

This parameter is meaningless unless BANNER=Y.
RTINFO2 Optional. Used for upload only. Specify any information which will assist in routing the printed report. This information will be printed on the pcMAINFRAME banner page for the report.

This parameter is meaningless unless BANNER=Y.

RETRY Optional. Used for download only. Specifies the number of retries that will be attempted if the spool access path is busy. If not specified, no retries will be made (numeric, 1-3 characters).

RETRYINT Optional. Used for download only. Specifies how long CFXJESEX will wait before retrying if a spool access busy condition occurs (numeric, 1-6 characters, form hhmmss).

This parameter is meaningless unless RETRY is specified.

TSLIMIT Optional. Used for download only. (numeric, 1-7 characters). If TSLIMIT is present, CFXJESEX will copy each report to CICS temporary storage before downloading. TSLIMIT specifies the maximum number of records CFXJESEX will write to temporary storage before abnormally terminating the download.

Sample Profiles

Included with your pcMAINFRAME installation material are six sample profiles which illustrate the most common types of spool access:

- **JES01** - Download a report from JES
- **JES02** - Upload a report to JES
- **JES03** - Upload binary file to JES
- **JES04** - Download binary file from JES
- **JES05** - Download a report with PC ASA carriage control
- **JES06** - Download a report with temporary storage queuing

**JES01** (report download) uses SPOOL format so that page breaks are preserved in the downloaded file. USERID is specified, as it always must be for downloads. CCTYPE is "A", indicating that the carriage control format is ASA. PURGE is set to "Y", which causes the JES dataset to be deleted when the download is completed. ALLFILES is "Y", which causes all JES data sets to be downloaded. If the spool facility is busy, ten retries will be made at 30 second intervals (RETRY is 10, RETRYINT is 30).

**JES02** (report upload) also uses SPOOL format. Here, DESTNODE and DESTUSER are included to identify the JES destination node and user. The uploaded report will be prefixed with a banner page (BANNER is "Y", and RTINFO1 and RTINFO2 are included). The uploaded report will utilize ASA control characters (CCTYPE is A); class B is used for the report.

**JES03** (binary upload) uses TRANSPARENT format to preserve hexadecimal data. Normally, we use this format only to transfer binary files for another user to download (nothing is to be printed); therefore, CCTYPE is set to "N" so no carriage control character is used.
DESTNODE and DESTUSER are included, as required. Class B is used.

**JES04** (binary download) again uses TRANSPARENT format. This profile would be appropriate for downloading a file uploaded with profile JES04. It includes only the USERID, CCTYPE, and CLASS exit parameters.

**JES05** (report download) uses SPOOL format; in addition, PCCTL is set to ASA so the downloaded PC file will contain ANSI-standard control characters in the first position of each record.

**JES06** (report download) includes the TSLIMIT specification; the report to be downloaded is queued to CICS temporary storage before the download begins.

*Upload/Download Specifications*

See "Upload/Download Specifications" under CFXPOWEX. The same considerations apply here, with one exception: for transparent data, the LRECL specification is irrelevant for **CFXJESEX** transfers.
Appendix F - CFX$FILE User Exit

CFX$FILE

CFX$FILE is a user exit coded by cfSOFTWARE that allows you to transfer data to and from the PC by way of CICS temporary storage queues.

As with all pcMAINFRAME transfers, you describe the upload and/or download of the data with a pcMAINFRAME profile. The profile identifies the name of the temporary storage queue and all other information.

1. Enter CFX$FILE as the Exit Name.
2. Enter EXIT as the File Type.
3. If the data to be uploaded/downloaded is report text and you wish to preserve the ASA carriage control, specify the Data Type as SPOOL. This data type of SPOOL should only be used with reports that have ASA carriage control.
4. Supply the required parameters on the User Exit Parameters profile screen. These parameters are described in the next section.

IND$FILE

Data uploaded or downloaded using the IBM facility named IND$FILE can be accessed via the CFX$FILE exit. The IBM standard for the naming of temporary storage queues used with IND$FILE is as follows:

1. The first four characters of the queue name are "CFTR".
2. The last four characters of the queue name are the terminal ID (EIBTRMID) for this CICS transaction.

See the "QNAME/HEADER Format" section for examples of how to specify this type of queue name to CFX$FILE.
### CFX$FILE Exit Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Length</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>QNAME</td>
<td>1 to 8</td>
<td>The 1 to 8 character name of the temporary storage queue to be used. There is no default QNAME; this parameter must be supplied in the exit parameters. Format: <code>[x..][=var][x...]</code> (see QNAME/HEADER Format below)</td>
</tr>
</tbody>
</table>
| DISP           | 1      | Disposition of queue after upload/download  
|                |        | R = re-use the queue <default>  
|                |        | download: R simply means RETAIN (keep the queue)  
|                |        | upload: R means delete the queue if it already exists and REUSE the queue name.  
|                |        | D = delete temporary storage queue  
|                |        | download: the queue will be deleted after the download completes.  
|                |        | upload: this disposition has no effect; the queue will still be retained after upload.  
|                |        | A = add-on to existing temporary storage queue  
|                |        | download: this disposition has no effect; the queue is retained after the download.  
|                |        | upload: this disposition means ADD the new data uploaded onto the end of the queue. If this disposition is used with a queue that does not exist, then the upload will simply create the new queue. |
| QTYPE          | 1      | Indicates whether the temporary storage queue is in memory (MAIN) or on disk (AUX).  
|                |        | M = MAIN memory for queue (default) (When CICS terminates, this queue will disappear).  
|                |        | A = AUX disk for queue (When CICS terminates, this queue will be retained for later access). |
| HEADER         | 1 to 25| The queue header record. The 1 to 25 character value used as the first 25 bytes in the queue header record. The queue header record is always 80 bytes long (blank filled after the 1st 25 bytes).  
|                |        | upload: this is the 1st record written to the temporary storage queue.  
|                |        | download: this parameter is ignored --- if a header was written to the queue, it will be downloaded whether or not this parameter is given.  
|                |        | Format: `[x..][=var][x...]` (see QNAME/HEADER Format below)  
|                |        | Note: If you are downloading to an application that previously used "Receive" to transfer data, you will get the IND$ header with pcMAINFRAME. To avoid this, set the "Skip before processing" value to 1 on the first profile screen. |
**QNAME/HEADER Format**

The QNAME may be from 1 to 8 characters long; the HEADER value may be from 1 to 25 characters long. Both will be left-aligned and blank-filled.

Certain [=var] (variables) may be specified as prefix, infix, or suffix. These variables are all 4-character symbols that are replaced by 4-character values as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Replacement Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>=TRM</td>
<td>EIBTRMID [CICS terminal ID for this pcMAINFRAME session]</td>
</tr>
<tr>
<td>=TAS</td>
<td>EIBTASKN [CICS task number for this pcMAINFRAME session (4 digit)].</td>
</tr>
<tr>
<td>=DAT</td>
<td>EIBDATE [CICS current Julian date in the form YDDD (year, day day day)]</td>
</tr>
<tr>
<td>=TIM</td>
<td>EIBTIME [CICS current time in the form HHMM (hour hour, minute minute)]</td>
</tr>
</tbody>
</table>

**Example 1:**

<table>
<thead>
<tr>
<th>Parm Name</th>
<th>Length</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>QNAME</td>
<td>8</td>
<td>CFTR=TRM</td>
</tr>
</tbody>
</table>

In the above example the QNAME is given as the characters "CFTR" followed by the EIBTRMID value. If the TRMID were "T100", then the QNAME would be "CFTRT100".

The CFTR=TRM QNAME is used by IBM IND$FILE formatted temporary storage queues.

**Example 2:**

<table>
<thead>
<tr>
<th>Parm name</th>
<th>Length</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEADER</td>
<td>25</td>
<td>CFTR=TRM =DAT=TIM</td>
</tr>
</tbody>
</table>

In the above example the HEADER would be valued as follows:

The 1st 8 bytes would be "CFTRT100" followed by a blank followed by YDDD of the current Julian date followed by HHMM of the current time.
CFX$FILE Sample Profile

Profile Name........ CFX$TXT Name of this definition
Profile Description.. CFX$FILE - FIXED TEXT WP UP/DNLOAD (NONE) If no password required
Password (optional) .. NONE A-Z, * for all classes
File or Exit name.... CFX$FILE FCT or FPT entry name
File type............ EXIT ESDS, KSDS, RDDS, Lib, Publ, Exit, User
Output data format... F FIXED B, D, F, T, S, W, X, I, L, A, V
Transfer direction... B (U)p, (D)own, (B)oth ways
Generate column headers (Y/N) .. N Translation Modes U, W, or M .. M
Read limit (Maximum reads) ... 0 Make blank fields Null (Y/N) .. N
Write limit (Maximum writes) .. 0 Multiple Record Header (Y/N) .. N
Skip before processing..... 0 Last maint/used dates: 02/28/98 02/28/98
Fixed rec length (optional) ..

Profile found - Update assumed.

PF1 - Return to Directory, PF2 - Next Page, Clear - Menu

Figure F-1: Profile Definition

In the above profile screen, the EXIT is defined as program CFX$FILE for both upload and download. The data format is Fixed-Text with Translation Mode mixed (upper and lower case kept) selected.

Parameter Maximum Data Dec Name Length Type Plc Parameter Value
QNAME 8 C CFTR=TRM DISP 1 C &DISP QTYPE 1 C M HEADER 25 C CFTR=TRM =DAT=TIM =TAS

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete.
PF1 - 1st page, PF2 - Next Page, PF3 - Prior Page, CLEAR - Menu.

Figure F-2: User Exit Parameters

In the above profile screen, the EXIT parameters are defined. Note: DISP was left as a User-Variable (&DISP) to be prompted for at execution time. Any of the parameters can be $variable. The QNAME will be formed of “CFTR” and the CICS terminal ID.
Define Include/Exclude rules...
from none to 12 are allowed.

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete.
Pf1 - 1st page, Pf2 - Next Page, Pf3 - Prior Page, Clear - Menu.

**Figure F-3: Record Selection**

Define rules for the break - If any rule is true the output break will occur.
From none to six rules are allowed.

Commands: >A - Insert after, >B - Insert before, = - Duplicate, < - Delete.
Pf1 - 1st page, Pf2 - Next Page, Pf3 - Prior Page, Clear - Menu.

**Figure F-4: Break Control**

In the above profile screens, the Record-Selection and Break-Control have been left null (all records selected and no break control). However, for download processing it is possible to use these facilities to select/summarize the records being returned from CFX$FILE.

VARIABLE, that is, the entire record for whatever length (up to a maximum of 4K) will be uploaded or downloaded.
pcMAINFRAME runs as a normal CICS or TSO application. Nothing in pcMAINFRAME circumvents your existing security. The standard pcMAINFRAME security functions of User and Profile classes can extend your existing security. Further enhancement of security comes in the form of a user written Security Exit program which provides administrators with greater flexibility in restricting access to specific pcMAINFRAME functions. This appendix describes the rules for coding a Security Exit and shows a sample exit.

When a Security Exit is defined on the Options Customization screen, pcMAINFRAME will invoke it via a CICS link, once for each PCMM menu function selected and for each transfer request. If the Security exit program returns a code other than OK, the message in SEC-MSG will be displayed and the user will not be allowed to continue.

Figure G-1 is a COBOL description of the control block used to communicate between the Security exit and pcMAINFRAME. A machine readable copy of the control block is contained in the source library that was shipped with your system under the name CFXSECXC.
The description below identifies the fields in the control block and describes their usage and any special consideration in using them.

**SEC-REQUEST** - this field is filled in by pcMAINFRAME and contains one of the following values:

- **S** - Signon. The user is beginning a transfer session. pcMAINFRAME will pass the transaction ID used to invoke the transfer, the PC ID being used during this transfer, and the terminal ID being used.

- **T** - Utility. The user is invoking the mainframe utility function during a transfer session. pcMAINFRAME will pass the transaction ID used to start the transfer, the PC ID, and the terminal ID.

- **U** - Upload. At the beginning of each upload, pcMAINFRAME invokes the security exit passing the TRANID, PCID, TERMID, name of the profile being used, and either the FCT of the file being uploaded to or the PPT of the exit program being used.

- **D** - Download. At the beginning of each download, pcMAINFRAME invokes the security exit passing the TRANID, PCID, TERMID, name of the profile being used, and either the FCT of the file being uploaded to or the PPT of the exit program being used.

- **1 through 8** - values 1 through 8 correspond to items on the PCMM main menu. Each time a user invokes an option from the menu the Security exit is invoked passing the transaction ID, and terminal ID being used.
The Options Customization screen is being invoked from the PCMM main menu. pcMAINFRAME passes the transaction ID, SEC-TRANID - contains the 4 character CICS transaction ID used to invoke that pcMAINFRAME function.

SEC-PCID - contains the 4 character pcMAINFRAME PC ID used during a transfer.

SEC-TERMID - contains the 4 character CICS terminal used being used.

SEC-FCT - contains the 8 character FCT name of the file being uploaded to, or downloaded from during a transfer.

SEC-PPT - contains the 8 character PPT name of the user exit programming being used.

SEC-PROFILE - contains the 8 character name of the profile being used during a transfer session.

SEC-RETURN-CODE - this field must be set by the exit with one of the following values to indicate the status of the processing:

- O - the user is authorized to perform the function, processing will continue.
- E - the user is not authorized to perform the function, terminate processing.

SEC-MSG - the exit may place a message here to be displayed on the user's screen indicating the reason for terminating the session.

Sample Security Exit

The following is an example of a coded security exit:

```
IDENTIFICATION DIVISION.
PROGRAM-ID. CFXSECX.
DESCRIPTION. PCMAINFRAME SAMPLE SECURITY EXIT PROGRAM.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
77 FILLER PIC X(45 VALUE '>>> COPYRIGHT (C) 1997 CF SOFTWARE, INC. <<<').
77 FILLER PIC X(35) VALUE 'CFXSECX V 4.00 07/25/91 13:30'.
01 DEFAULT-DATA-AREA.
  * Allowed PCID............
    05 DEF-DATA.
      10 DEF-PCIDS.
        15 FILLER PIC X(04) VALUE 'DEMO'.
        15 FILLER PIC X(04) VALUE 'CHB'.
      10 DEF-PCID-WORK REDEFINES DEF-PCIDS.
      15 DEF-PCID-TABLE OCCURS 2 TIMES INDEXED BY PCID-IX.
      20 DEF-PCID PIC X(04).
  * Allowed TERMID............
    10 DEF-TERMIDS.
      15 FILLER PIC X(04) VALUE 'D204'.
      15 FILLER PIC X(04) VALUE 'D205'.
      15 FILLER PIC X(04) VALUE 'D202'.
      15 FILLER PIC X(04) VALUE 'D203'.
    10 DEF-TERMID-WORK REDEFINES DEF-TERMIDS.
    15 DEF-TERMID-TABLE OCCURS 4 TIMES INDEXED BY TERMID-IX.
    20 DEF-TERMID PIC X(04).
```
* Allowed FCT members........
  10 DEF-FCTS.
    15 FILLER PIC X(08) VALUE 'CFXLIB1'.
    15 FILLER PIC X(08) VALUE 'CFXLIB2'.
    15 FILLER PIC X(08) VALUE 'CFXKSDS'.
    15 FILLER PIC X(08) VALUE 'CFXSSSD'.
  10 DEF-FCT-WORK REDEFINES DEF-FCTS.
   15 DEF-FCT-TABLE OCCURS 4 TIMES
       INDEXED BY FCT-IX.
      20 DEF-FCT PIC X(08).

* Allowed PPT members........
  10 DEF-PPTS.
    15 FILLER PIC X(08) VALUE 'CFXPOWX'.
    15 FILLER PIC X(08) VALUE 'CFXEXIT1'.
    15 FILLER PIC X(08) VALUE 'CFXEXIT2'.
    15 FILLER PIC X(08) VALUE 'CFXEXIT3'.
    15 FILLER PIC X(08) VALUE 'CFXTST1'.
    15 FILLER PIC X(08) VALUE 'CFXTST2'.
    15 FILLER PIC X(08) VALUE 'CFXTST3'.
  10 DEF-PPT-WORK REDEFINES DEF-PPTS.
   15 DEF-PPT-TABLE OCCURS 7 TIMES
       INDEXED BY PPT-IX.
      20 DEF-PPT PIC X(08).

* Allowed PROFILE under certain PCID...
  10 DEF-PROFILE-NAMES.
    15 FILLER PIC X(08) VALUE 'PCMT007'.
    15 FILLER PIC X(08) VALUE 'LIB'.
    15 FILLER PIC X(08) VALUE 'LIBT'.
    15 FILLER PIC X(08) VALUE 'SALES1'.
    15 FILLER PIC X(08) VALUE 'DEMO9'.
    15 FILLER PIC X(08) VALUE 'POWER01'.
    15 FILLER PIC X(08) VALUE 'POWER04'.
    15 FILLER PIC X(08) VALUE 'POWER07'.
  10 DEF-PROFILE-NAME-WORK REDEFINES DEF-PROFILE-NAMES.
   15 DEF-PROFILE-NAME-TABLE OCCURS 8 TIMES
       INDEXED BY PROFILE-IX.
      20 DEF-PROFILE-NAME PIC X(08).

LINKAGE SECTION.
 01 DFHCOMMAREA.
COPY CFXSECXC.
PROCEDURE DIVISION.

0000-MAINLINE.
  PERFORM 0100-HOUSEKEEPING THRU 0100-HOUSEKEEPING-EXIT.
  IF SEC-TRANSFER-SIGNON
      PERFORM 0200-SIGNON-CHECK THRU 0200-SIGNON-CHECK-EXIT
  ELSE
      IF SEC-TRANSFER
          PERFORM 0300-TRANSFER-CHECK THRU 0300-TRANSFER-CHECK-EXIT
      ELSE
          IF SEC-PROFILE-MAINT OR SEC-PCID-MAINT OR
          SEC-USER-PROFILE-MAINT
              PERFORM 0400-MAINT-CHECK THRU 0400-MAINT-CHECK-EXIT
          ELSE
              IF SEC-OPTIONS-CUSTOMIZATION
                  PERFORM 0500-CUST-CHECK THRU 0500-CUST-CHECK-EXIT
              ELSE
                  IF SEC-LIBRARY-DIRECTORY OR
                  SEC-PROFILE-DIRECTORY OR
                  SEC-PCID-DIRECTORY
                      PERFORM 0600-DIR-CHECK THRU 0600-DIR-CHECK-EXIT
                  ELSE
                      IF SEC-REVIEW-TRAN-LOG
                          PERFORM 0700-TRAN-LOG-CHECK THRU 0700-TRAN-LOG-CHECK-EXIT
                      ELSE
                          IF SEC-USER-PROFILE-DIRECTORY
                              PERFORM 0800-USER-CHECK THRU 0800-USER-CHECK-EXIT.
                      END-IF
                  END-IF
              END-IF
          END-IF
      END-IF
  END-IF
  EXEC CICS
  RETURN
END-EXEC.
0000-MAINLINE-EXIT.
EXIT.

0100-HOUSEKEEPING.
* Initialize the message area..............
MOVE LOW-VALUES TO SEC-MSG.
MOVE '0' TO SEC-RETURN-CODE.

0100-HOUSEKEEPING-EXIT.
EXIT.

0200-SIGNON-CHECK.
IF SEC-PCID > LOW-VALUE
SEARCH DEF-PCID-TABLE
AT END
MOVE 'E' TO SEC-RETURN-CODE
MOVE 'PCID NOT VALID' TO SEC-MSG
WHEN DEF-PCID (PCID-IX) = SEC-PCID
NEXT SENTENCE.

0200-SIGNON-CHECK-EXIT.
EXIT.

0300-TRANSFER-CHECK.
IF SEC-TRANSFER-UTILITY
IF SEC-TERMID = 'D205'
MOVE 'E' TO SEC-RETURN-CODE
MOVE 'UTILITY OPTION CAN NOT BE ACCESSED' TO SEC-MSG.

IF NOT SEC-TRANSFER-UTILITY
SEARCH DEF-PROFILE-NAME-TABLE
AT END
MOVE 'E' TO SEC-RETURN-CODE
MOVE 'USE OF PROFILE IS NOT VALID' TO SEC-MSG
GO TO 0300-TRANSFER-CHECK-EXIT
WHEN DEF-PROFILE-NAME (PROFILE-IX) = SEC-PROFILE-NAME
NEXT SENTENCE.

IF NOT SEC-TRANSFER-UTILITY
SEARCH DEF-FCT-TABLE
AT END
MOVE 'E' TO SEC-RETURN-CODE
MOVE 'TRANSFER NOT ALLOW TO THIS FILE' TO SEC-MSG
GO TO 0300-TRANSFER-CHECK-EXIT
WHEN DEF-FCT (FCT-IX) = SEC-FCT
NEXT SENTENCE.

IF NOT SEC-TRANSFER-UTILITY
IF SEC-PCID = 'DEMO' AND SEC-PROFILE-NAME NOT = 'SALES1'
AND SEC-TERMID = 'D205'
MOVE 'E' TO SEC-RETURN-CODE
MOVE 'PCID NOT ALLOWED TO USE THIS PROFILE' TO SEC-MSG.

0300-TRANSFER-CHECK-EXIT.
EXIT.

0400-MAINT-CHECK.
IF SEC-TERMID = 'D205'
MOVE 'E' TO SEC-RETURN-CODE
MOVE 'CHANGES ARE NOT ALLOWED' TO SEC-MSG.

IF SEC-PCID-MAINT
IF SEC-TERMID = 'D205'
MOVE 'E' TO SEC-RETURN-CODE
MOVE 'PCID CAN NOT BE UPDATED' TO SEC-MSG.

0400-MAINT-CHECK-EXIT.
EXIT.

0500-CUST-CHECK.
IF SEC-TERMID = 'D205'
MOVE 'E' TO SEC-RETURN-CODE
MOVE 'OPTION ARE NOT ALLOWED TO CHANGE' TO SEC-MSG.

0500-CUST-CHECK-EXIT.
EXIT.
0600-DIR-CHECK.
   MOVE 'E' TO SEC-RETURN-CODE.
   MOVE 'DIRECTORY CAN NOT BE ACCESS' TO SEC-MSG.
   0600-DIR-CHECK-EXIT.
   EXIT.

0700-TRAN-LOG-CHECK.
   MOVE 'E' TO SEC-RETURN-CODE.
   MOVE 'VIEW TRAN LOG ARE NOT ALLOWED' TO SEC-MSG.
   0700-TRAN-LOG-CHECK-EXIT.
   EXIT.

0800-USER-CHECK.
   MOVE 'E' TO SEC-RETURN-CODE.
   MOVE 'USER OPTION CAN NOT BE ACCESS' TO SEC-MSG.
   0800-USER-CHECK-EXIT.
   GOBACK.

*Figure G-2: Sample Security Exit*
Appendix H - BIM-Edit and VSE Library Access

CFXBIMX is a user exit coded by cfSOFTWARE that allows you to transfer data between your PC and your mainframe’s BIM-EDIT library or DOS/VSE sublibrary. To use CFXBIMX, your mainframe must have installed BIM-EDIT software and must be running DOS/VSE SP version 2.1 or above.

As with other types of file transfer, you must describe your requirements in a pcMAINFRAME profile. The profile parameters are the same as for any other type of transfer, but you must also:

- Enter CFXBIMX as the Exit name and EXIT as the File Type on the first Profile Definition screen;
- Supply the required parameters on the “User Exit Parameters” Profile Definition screen. These parameters define the BIM-EDIT library or DOS/VSE sublibrary, Logon ID, Password, and other related information required to accomplish the transfer.

For DOS VSE and BIM-EDIT transfers, the maximum record size allowable is 253 bytes. If larger records occur in the data, they are truncated.

Any pcMAINFRAME supported PC file format, including Fixed, Excel, dBASE, Transparent, etc. can be user with CFXBIMX. A BIM-EDIT User and Password must be supplied for each file transfer. This ID is used to define the user to BIM-EDIT. All of BIM-EDIT’s library access rules for the user defined are applied during data transfer.

CFXBIMX Exit Parms for Data Transfers

Common Parameters

**BIMID Optional.** The BIM-EDIT System Name defined during BIM installation. If this parameter is not supplied, BIMEDIT is the default (1-8 characters).

**LOGON Required.** User Logon ID for BIM-EDIT system (1-8 characters).
PASSWORD Required. Password for User Logon-ID. (1-8 characters).

LIBTYPE Optional. The type of library to be accessed.
- **BIM** - a BIM-EDIT library
- **VSE** - a VSE sublibrary.

If this parameter is not supplied, ‘BIM’ is the default (3 characters).

LIBRARY Optional. The name of the library to be accessed. If this parameter is not supplied, the current BIM-EDIT library or VSE sublibrary will be used. (1 to 16 characters).

MEMBER Optional. The name of the library member for file transfer. If this parameter is not supplied, the PCFILE name defined in the setup file will be used. (1-8 characters).

**Upload only Parameters**

The following parameters are passed on to BIM-EDIT during uploads. They are fully documented in the BIM-EDIT User Reference Manual and function as documented there.

If an uploaded member is being added to the specified library, the parameters defined will be used in a DEFINE/DEFINED command. If the member already exists and REPLACE is specified, an ALTER command is issued after a successful upload. Attributes will NOT be changed if an existing member is being APPENDED to, or during CHECKIN.

**TYPE Optional for BIM; Required for VSE sublibraries.** Used to determine the initial attributes or the member. Types defined by BIM-EDIT are:

- **$DFL** Default template
- **ASM** Assembler program Source
- **COBOL** COBOL program Source
- **DATA** Data
- **FORT** FORTRAN program Source
- **JCL** DOS job streams
- **PLI** PLI program Source
- **PROC** Procedure
- **RPG** RPG program Source
- **TEXT** Textual material

**TITLE Optional (BIM only).** A comment or description of the member.

During automated upload, the LIBNOTE= entry of the PC’s setup file can be used to specify this parm. (1-32 characters)

**ATTR Optional (BIM only).** A site defined attribute typically used by the COMPILe command to determine the nature of the compile. See your BIM administrator.

**AUDIT Optional (BIM only) Specify ‘Y’ if auditing is desired. (1 character)**

**CASE Optional (BIM only) Specify ‘U’ for uppercase translation, ‘M’ for mixed case. If this parameter is not supplied, ‘U’ is the default.**
Note: there is a ‘translation mode’ field in the first Profile Definition screen that controls the ‘any case’ translation during file transfer. If mixed case transfers are desired, both fields must be set. (1 character)

CHECK Optional (BIM only) Specify if a member is to be updated only under checkin/checkout control. Specify ‘Y’ to set check control on. (1 character).

LIBDEF Optional (BIM only). A site defined parameter typically used by the COMPILE command to determine how to build compile JCL. See your BIM administrator.

USER Optional (BIM only). A purely documentary field that can be set to any value. During member create, USER is set to the logon-ID if this parameter is not valued. (1 to 8 characters).

STAMP Optional (BIM only). Specify ‘Y’ if member stamping is required. This parameter may be permanently turned on by your BIM administrator.

DATA Optional (VSE only). Specify ‘Y’ if a ‘PROC’ type VSE sublibrary member contains SYSIPT DATA. (1 character).

**Download only Parameters**

The following parameters are passed on to BIM-EDIT during downloads. They are fully documented in the BIM-EDIT System Reference Manual and function as documented there. These parameters are applied during standard download and when a member is being downloaded as part of a checkout request.

INCL Optional. Specify ‘Y’ if INCLUDES are to be expanded. (1 Character)

HEX Optional. Specify ‘Y’ if HEX replacement is to take place. Use of this parameter during download could be problematic since the transfer process converts EBCDIC to ASCII. Non-text characters cannot be validly converted and are transferred as spaces (UpperCase or Mixed translation mode), or sent ‘as-is’ (Word Process translation mode).

Use the default HEX=N to preserve the Esc-hex-hex sequence in the member during download/upload. If the HEX data is needed on the PC, a special translation table may be needed to define the desired EBCDIC to ASCII conversion. (1 Character)

VBL Optional. Specify ‘Y’ if variables in the member are to be replaced. (1 Character)

TRAP Optional. Specify ‘Y’ if lines in the member starting with the ‘text following trap escape character’ are to be processed and downloaded. The default value for this character is the right parenthesis “)”, although this may be changed by your BIM administrator. (1 Character).

PAD Optional. Used for download only. The record will be padded with spaces after extract from BIM member or VSE library member and before record select and extract rules are applied. The maximum length can be padded is 253. The default is no PAD. If physical record length is less than the maximum length in profile, use the PAD parameter. (numeric 3 bytes).
Download with Checkout

The CHECKOUT parm can be used create a slave copy of a member, then download the copy for update at the PC. CHECKOUT can only be used if the member specified is enabled for checkin/checkout (CHECK=ON), if the user has list access to the member’s library, and the member is not already checked out.

The following parameters are used to CHECKOUT and download a member.

- **LIBRARY** - The name of the library where the master member is located. If this parameter is not supplied, the current BIM-EDIT library is assumed.
- **MEMBER** - The name of the member to be checked out. If this parameter is not supplied, the PC file name in the setup file will be used. (1-16 characters).
- **CHECKOUT** - Specify ‘Y’ to request that a member be checked out. (1 character)
- **CHECKOUTLIB** - Required when CHECKOUT is set to ‘Y’. Specify the name of the library where the slave member is to be copied. The slave member will be created with all of the master member attributes and will be flagged as the slave in the checkout relationship. After the download, the member will remain in this library until CHECKIN.

Note: CHECKOUT is supported for BIM libraries only.

Upload with CHECKIN

The CHECKIN parameter can be used upload/update a slave copy of a member, then update the master copy and reset the checkout status. Only the valid slave copy of member can be used for CHECKIN.

A new member cannot be created during checkin. You must either update the member or append the data to the existing work copy. The attributes of the work copy will not be changed.

The following parameters are used to upload and CHECKIN a member.

- **MEMBER** - The name of the member to be checked in. If this parameter is not supplied, the PC file name in the setup file will be used. (1-16 characters).
- **CHECKIN** - Specify ‘Y’ to request that the member be checked in. (1 character)
- **CHECKINLIB** - During CHECKIN, specify the name of the library where the slave member resides. The upload will replace the member in this library before checkin. After a successful CHECKIN, the member will be deleted from this library. If this parameter is not supplied, the current BIM-EDIT library is assumed

Note: CHECKIN is supported for BIM libraries only.
Glossary of Terms

The words defined below are terms used throughout this manual that have special meaning in the pcMAINFRAME data processing environment.

3101
An asynchronous terminal manufactured by IBM. A terminal type commonly supported by protocol converters.

3270/4/6/8
A member of a family of terminal and supporting controllers manufactured by IBM. One way in which pcMAINFRAME communicate to the host is through 3270 emulation devices.

ASCII
1. USA Standard Code for Information Interchange. This is the code used to define transmitted characters in Asynchronous Communications. Most micro computers use this code for internal data representation.
2. A pcMAINFRAME data format in which records will be stripped of any trailing blanks. The variable length records will be written with CRLF characters as record terminators.

Asynchronous
A communications protocol generally used by low speed devices (110 - 19200 baud). This is the protocol used by pcMAINFRAME for dial up communications with protocol converters.

Average
One of the summarization functions performed by pcMAINFRAME in which the mean or average amount of a numeric field within a group of detail records is downloaded to a PC in place of the detail.

BASIC
One of several data formats provided by pcMAINFRAME also known as comma delimited. Data downloaded from the mainframe may optionally be converted into this format. This is the data format commonly used by programs written in the BASIC language and is characterized by variable length fields delimited by commas,
character data bracketed by quotation marks (") and records delimited by carriage return characters (CR) -see definition below.

**BASIC**
A programming language commonly used on micro and mini computers -see definition above.

**BAT File**
"Batch file", a text file on the PC that contains commands. The name of a BAT file may be entered as a command and PC DOS will execute all the commands within the BAT file.

**Baud**
BPS, a measure of speed for communications activity between computers and/or peripheral devices. A baud is a transmission rate of 1 bit or binary digit of information per second. It generally takes 7 - 10 bits to represent 1 character of information during transmission so that the number of characters transmitted per second is the baud divided by 10, i.e. 1200 baud = 1200/10 = 120 character per second.

**BISYNC**
or Binary Synchronous is a communications protocol used by higher speed devices and is a block protocol. pcMAINFRAME uses BISYNC communications indirectly when it is operated on a 3274/3276 remote control unit as a 3270 device.

**Break**
A change in data contained in a control field. This is an optional facility used by pcMAINFRAME to group detail records that are alike and produce a single summarized output record.

**CICS**
A telecommunications monitor marketed by IBM that provides a host facility for terminal oriented applications. pcMAINFRAME operates as a CICS application.

**COM**
This is a suffix appended to executable program files on the PC. These programs are executed by entering the file name on the DOS command line.

**Com port**
This is a communications port on the PC. It is used for asynchronous data transmission. Comports COM1 and COM2 are available on most PCs. pcMAINFRAME may use either.

**Compression**
A compaction of data performed before transmission to reduce the number of characters sent between pcMAINFRAME on the mainframe and on the PC and correspondingly reducing the transmission time required.
CR
Carriage Return, hex 0D. This is a code used in ASCII text files on the PC to delineate the end of a record. It is entered on the keyboard by using the \\ or \\ key.

CUT
Control Unit Terminal. A description of a 3270 terminal that has a single host session available to it.

DCT
Destination Control Table. This is a facility within CICS that describes external sequential files that may be accessed. pcMAINFRAME requires this facility to do hardcopy logging of activity. A destination is generally established to a printer for this log.

DFT
Distributed Function Terminal. A description of a 3270 terminal that has multiple host sessions available to it and performs many functions that are done by the communications controller in CUT mode.

DIF
Data Interchange Format. Most popular spreadsheet programs and some other applications such as databases use this format of data representation. This is one of the several data formats provided by pcMAINFRAME. Data being downloaded to a PC may optionally be converted into this format.

Disk
A magnetic storage device capable of randomly storing and retrieving large amounts of information. Disks are the most common storage device on most mainframes and small versions are available on PCs and are commonly called "hard-disks".

Diskette
A random access magnetic storage device commonly called a "floppy disk" that is the most common storage medium on most PCs.

DOS
Disk Operating System. This is a group of programs that manage the computer and provide service functions to other programs and applications. pcMAINFRAME requires PC DOS or MS DOS Release 3.1 or higher in order to function. These DOS programs are supplied by IBM and Microsoft respectively.

Download
A function performed by pcMAINFRAME that transfers data from a mainframe file to a PC file.

EBCDIC
Extended Binary Coded Decimal Interchange Code. It is a code used on IBM mainframes for internal representation of character information.
Emulator Card
An add-on interface card for the PC. This card allows a PC to emulate an IBM 3278 terminal and is used by pcMAINFRAME to support control unit attachment.

Error Level
This is a facility of PC DOS that allows a program to post a result code upon termination. Other programs or BAT files may test this result code and take appropriate action. pcMAINFRAME posts the error level upon completion, giving an indication of success or type of failure if one occurred.

ESC
The \[\text{ESC}\] key on the PC keyboard is used to inform pcMAINFRAME that the current function should be terminated. When this key is pressed, pcMAINFRAME stops any transfers currently in progress and returns to the menu.

ESDS
Entry Sequenced Data Set. This is a type of VSAM file in which records are added to the file sequentially with the last record in the file always being the last record added.

Excel
A PC data format supported by pcMAINFRAME. It is used by the Microsoft Excel Spreadsheet program. pcMAINFRAME can upload from Excel files and create Excel files on download.

EXE
This is a suffix appended to executable program files on the PC. These programs are executed by entering the file name on the DOS command line.

EXIT
A facility allowing user written CICS programs to interface with pcMAINFRAME. This provides a method of customizing and extending the functions of pcMAINFRAME.

FCT
File Control Table. This is a facility within CICS that describes the files that may be accessed. pcMAINFRAME requires that mainframe files to be accessed by PCs have an entry in the FCT. Files may be identified in profiles before they are placed in the FCT, but a warning message will be given.

File
A named, unique aggregate of information stored on a disk on either the mainframe or PC. pcMAINFRAME deals with files on both the mainframe and the PC.

Filename
This is the term used by PC DOS and MS DOS to refer to the name associated with a file. It contains an optional disk prefix with the filename consisting of a 1 - 8 character name and an optional 1 - 3 character suffix, i.e. B:\PAYROLL.DIF
Fixed
One of the data formats supported by pcMAINFRAME. It is characterized by fixed length fields within fixed length records that contain no field delimiters or non-data characters.

Gateway
A hardware and software communication facility for local area networks. Gateways allow multiple PCs on a network to access the host (generally as a 3270 terminal) through a single physical connection. pcMAINFRAME supports file transfer through most gateway products.

Include
A pcMAINFRAME data format that permits multiple files to be uploaded as a single operation. The user may specify the order in which the files are put into the uploaded data stream.

Insert
This is a download function of pcMAINFRAME. Insert places constant text into the output record being downloaded.

KSDS
Keyed Sequenced Data Set. This is a type of VSAM file in which each record has a key associated with it and the file is organized in ascending order of keys. Records may be retrieved directly by key or read sequentially.

LAN
Local Area Network. A method of linking PCs and other devices together to share data. This term applies to a variety of hardware methods of connecting the devices. pcMAINFRAME operates over most local area networks. PCs on these networks can upload and download data through gateways to the host.

Last
This is a download function of pcMAINFRAME. Last causes the last value of a data field to be sent to the PC when a break occurs.

LF
Line feed, hex 0A. This code is used in ASCII text files in conjunction with a carriage return to delineate the end of a record.

Library
This is a special mainframe file used by pcMAINFRAME to support storage of PC uploaded data. Files are stored within the library by their PC filename.

Modem
Communications equipment used to attach computers or terminals to a telephone line so that they may communicate. Modems are required on both the PC and the mainframe when pcMAINFRAME is used to establish a remote dial-up connection.
**Move**
This is a download function of pcMAINFRAME. Move causes the last value of a data field to be sent to the PC when a break occurs. Move and Last are synonymous.

**PA Keys**
Program Attention Keys. These are control function keys on 3270 type terminals.

**Packed**
A numeric data format commonly used in mainframe files. It saves space and allows direct use in computations but is generally unsupported on micro computers. pcMAINFRAME automatically converts packed data to a format usable on the PC. To maintain its packed format when downloaded, fields may be defined as transparent.

**PC**
Personal Computer: in the context of pcMAINFRAME, this means an IBM PC, IBM PC/XT, IBM PC/AT, IBM PS/2 or a fully compatible computer.

**PC ID**
A unique four character identifier associated with each PC using the system.

**PF Keys**
Program Function Keys. These are control keys on a 3270 type terminal and are used by pcMAINFRAME for non-transmission functions.

**Profile**
This is a description of data that is to be transferred by pcMAINFRAME. It describes the file, data type, record and field selection, and output format of data that is transferred. PC users access the data by profile name.

**Protocol Converter**
A device that converts one type of communications protocol into another. Within the context of pcMAINFRAME it is a device that communicates with a PC or a terminal in asynchronous protocol and communicates with a mainframe in binary synchronous or SDLC protocol or is channel attached making the PC or terminal appear to be a 3270 type terminal. pcMAINFRAME supports communications through most protocol converters.

**Return**
A key on the PC keyboard that functions to inform pcMAINFRAME that input for a particular field is complete. It is used in a similar fashion to the (Enter) key on a 3270.

**Screen**
The display appearing on either a terminal or a PC.
**Sequence**
An output operation of pcMAINFRAME which places a sequential number into the output record.

**Setup file**
A file maintained on the PC that contains the necessary information to automatically establish communication with the mainframe and perform any necessary logging-on functions.

**Spool**
A data type used by pcMAINFRAME that represents print formatted information. It contains printer control characters such as line spacing and top of forms. These characters are transmitted correctly between the PC and mainframe.

**Sum**
One of the summarization functions performed by pcMAINFRAME in which the total or sum of a numeric field within a group of detail records is downloaded to a PC in place of the detail.

**Terminal Emulation**
Software resident on the PC that causes it to behave as if it were a 3270 type terminal. pcMAINFRAME and Dialog provide this facility.

**Time-Out**
An error condition occurring during transmission in which either the mainframe or the PC fails to receive a response within the appropriate period of time.

**Transmission**
The process of communication between the PC and the mainframe. Transmissions consist of messages that are passed back and forth containing data, control information, and messages.

**Transparent**
A data format provided by pcMAINFRAME as an option to allow sending non-character files such as programs, word processor work files, and spreadsheet models between PCs and the mainframe.

**Transparent field**
An output format data type that allows pcMAINFRAME to send non-character data at a field level.

**Upload**
A function performed by pcMAINFRAME that transfers data from a PC file to a mainframe file or the PC’s library.
**Variance**
One of the summarization functions performed by pcMAINFRAME in which the statistical function variance is calculated on a numeric field within a group of detail records and downloaded to a PC in place of the detail.

**VSAM**
Virtual Storage Access Method. A method of storing and retrieving information on disk files in common use on large IBM computers.

**VT100**
An asynchronous terminal manufactured by Digital Equipment Corporation. A terminal type commonly supported by protocol converters.

**WKS**
A pcMAINFRAME data format that reads and writes Lotus WK1 format files.

**xBASE**
A pcMAINFRAME data format that reads and writes dBASE III and dBASE IV format files.
## Index

### 1
19xx cut off year ............................................. 55, 56, 82

### 3
3270 Block Size .................................................. 82

### A
Abend ........................................................................ 82
Access Ranges and Limits ........................................ 17, 42–43, 68
Administration .......................................................... 73–83
Backup and Recovery ............................................... 74
Centralized Control ................................................... 8
Logging ..................................................................... 75
Options Customization ............................................. 80–83
Profile Definitions ................................................... 75–76
Profile Reports ....................................................... 76–79
Security ...................................................................... 73–74
ASCII Format ................................................................ 37

### B
Backup and Recovery .................................................. 74
Basic Format ................................................................ 37
Date Expansion ....................................................... 56
Downloads from POWER Queue ................................ 172
Uploads to POWER Queue ........................................ 171
Batch Transfers ...................................................... see Host Initiation
Binary Format .......................................................... 38
Blanks ....................................................................... see Hex Data Entry
Block Size ............................................................... 82, 97, 128, 129
Break Control ........................................................... 6, 52–53
Browse ....................................................................... 6, 76, see also Profile
Btrieve File Access
  Field Type Definition ............................................... 136
  Format ..................................................................... 37
  Mainframe Requirements ....................................... 135–36
  PC Requirements ..................................................... 137

### C
Centralized Control .................................................... 8
CFXSFILE ............................................................... 179–84
  Parameters for ......................................................... 180–81
  Sample Profiles ....................................................... 182–84
CFXBIMX ............................................................... 191–94
  Parameters
    Checkin/Checkout ............................................... 194
    Download ........................................................... 193
    Upload ................................................................... 192–93
CFXFORM, Backup and Recovery ................................ 75
CFXINIT ................................................................. see Host Initiation
CFXESEX ................................................................. 173–78
  Download Bottlenecks and ....................................... 175
  Downloading Reports .............................................. 174
  Parameters for ......................................................... 175–77
  Sample Profiles ....................................................... 177–78
  Uploading Reports ................................................... 175
  Uploads and Downloads ........................................... 178
CFXLIB, Backup and Recovery .................................... 75
CFXLIBU ................................................................. 123–34
  Adding a Member ................................................... 132
  Appending Data to a Member ................................... 133
  Command File ......................................................... 124–30
    Parameter Descriptions ......................................... 127–30
    Return Codes ....................................................... 125
    Syntax ................................................................... 124–25
Examples
  MVS ................................................................. 134
  VSE ..................................................................... 131–33
JCL Requirements .................................................... 102, 131
  Retrieving a Member .............................................. 133
  Setup Command Syntax .......................................... 133
CFXPOWERX ........................................................ 165–72
  Command Processing Parameters ............................ 169
  Commands ............................................................ 172
  Download Specifications .......................................... 171–72

Bulk File Transfer .................................................. see Host Initiation
Generic Transfers ........................................... 169
Parameters ................................................ 166–69
Upload Specifications .................................... 170–71
Character Translation .................................... 139–48
Customization .................................................. 146–47
Language Specific ............................................ 141–46
U.S. English ..................................................... 141
Spool Control Characters ................................ 147
Standard Characters ...................................... 139–40
CICS Security .................................................. 74
Class Codes .................................................. 36, 62, 67, 80
Specifying in Master Profiles ....................... 67
Clipper .......................................................... see xBASE
CLIST
PCMM .............................................................. 24, 31
Component Summary ..................................... 4–5
Customization of pcMAINFRAME
19xx cutoff year .............................................. 82
3270 Block Size .............................................. 82
Activity Logging ............................................. 81
CICS Abend ................................................... 82
Class Codes .................................................. 80
Exit Support .................................................. 82
History Logging ............................................. 81
Individual Form File ...................................... 81
Log Destination ............................................ 81
Message and Screen Language ...................... 82
Protocol Converter Block Size ...................... 82
Security Exit Name ......................................... 81
Syncpoint Record Limit .................................. 82

D
Data Formatting .............................................. 5, 6, 69. see also Profile
For User Profile ........................................... 69
Data Integrity .................................................. 9
Data Selection .............................................. 6–7
Data Types Supported
Defining ....................................................... 54–57
Mainframe ...................................................... 54–56
PC ................................................................. 37–38
Date Formats .................................................. see Profile
dBASE .......................................................... see xBASE
DDREC .......................................................... 24, 37, 64
Add ............................................................... 28
Dataset Information ....................................... 71
DCB Information .......................................... 72
Delete ............................................................ 29
Maintenance ................................................. 28, 70–72
Update .......................................................... 29
DDREC Directory ........................................... 27–28, 72
DIF Format .................................................... 37, 171, 172
DL/I ................................................................. 158
Download ..................................................... 5, 39, 54
From Multiple Files see Profile, Multiple File
Definition
of xBASE file .................................................. 54
Record Selection ........................................... 47–49

E
End Table .................................................. see Tables
Error Recovery .............................................. 9
Excel ............................................................. 37
Excel Format .................................................. 37, 55, 56, 57
Exiting pcMAINFRAME .................................... 72

F
Features .................................................. 5–9. See also entry for specific feature
Central Control ............................................. 8
Data Formatting ............................................ 6
Data Selection .............................................. 6–7
Library Support ............................................. 8
Security Exit .................................................. 6
User Exit ...................................................... 5
User Profile ................................................... 7
Field Format .................................................. 48, 52
Field Order ................................................... 53
File Types Supported
Mainframe ..................................................... 5
PC ................................................................. 5, 7
Fixed Format .............................................. 37, 39, 171, 172
Date Expansion ............................................ 56
Downloads from POWER Queue .................. 172
Specifying Record Length ......................... 39
Uploads to POWER Queue ......................... 171
Fixed Table .................................................. see Tables
Form file
Individual ................................................... 24, 62, 81
System ......................................................... 24, 81
Foxbase .......................................................... see xBASE
Function Menu ............................................ see xBASE
Functional Walk-through ............................ 12–23
Functional Walk-through TSO ...................... 24–29

G
Generic Transfers
Library ....................................................... 116
User Exit ..................................................... 169

H
Hex Data Entry ............................................. 43
Host Initiation ............................................. 85–114
3270 Connect .............................................. 93
Batch Transfer ............................................. 86–88
Command File
<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>55</td>
</tr>
<tr>
<td>56</td>
</tr>
<tr>
<td>57</td>
</tr>
<tr>
<td>38</td>
</tr>
<tr>
<td>202</td>
</tr>
<tr>
<td>33</td>
</tr>
<tr>
<td>102</td>
</tr>
<tr>
<td>93</td>
</tr>
<tr>
<td>108-13</td>
</tr>
<tr>
<td>96</td>
</tr>
<tr>
<td>100-101</td>
</tr>
<tr>
<td>89-90</td>
</tr>
<tr>
<td>93</td>
</tr>
<tr>
<td>103-8</td>
</tr>
<tr>
<td>161</td>
</tr>
<tr>
<td>38</td>
</tr>
<tr>
<td>102</td>
</tr>
<tr>
<td>82</td>
</tr>
<tr>
<td>112</td>
</tr>
<tr>
<td>116</td>
</tr>
<tr>
<td>117</td>
</tr>
<tr>
<td>119-23</td>
</tr>
<tr>
<td>74</td>
</tr>
<tr>
<td>62, 74, 74</td>
</tr>
<tr>
<td>115</td>
</tr>
<tr>
<td>119</td>
</tr>
<tr>
<td>63, 74, 117</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>87</td>
</tr>
<tr>
<td>33, 81</td>
</tr>
<tr>
<td>81</td>
</tr>
<tr>
<td>93</td>
</tr>
<tr>
<td>55, 56, 57</td>
</tr>
</tbody>
</table>

M
- Master Profile
- Maximum Record Length
- Multiple File Support
- Multiple Record Files

O
- Occurs
- Online Functions
- Online Directory
- Overview of Processing

P
- Password
- PC ID
- Authorized Prefix
- Creation
- Customized Save
- Customized View
- Library Assignment
- Logging
- Maintenance
- TSO
- Update
- Updating from the directory
- User Profile Customization Limits

System Administrator’s Guide

Index ● 205
Form File .................. 24, 62, 81
Function Walk-through .... 24–29
Logging ......................... 33
Options Customization ...... 80
PDS ............................ 37
Profile Directory ............... 27
QSAM ........................... 37
Transfer File Name ............ 37
User Exits ..................... 149
User ID .................... 4, 25–27. see also User ID
User ID Maintenance ......... 61–63

U
Upload ......................... 5, 38
User Exit .................. 43, 82, 173–78, 185–90
Accessing a Database Example 164
and Profiles .................. 149
CFXXFILE ...................... 179–84
Parameters for .................. 180–81
Sample Profiles .............. 182–84
CFXBIMX ...................... 191–94
Parameters
Checkin/Checkout .............. 194
Download .................... 193
Upload .................. 192–93
CFXJESEX ...................... 173–78
Download Bottlenecks and ... 175
Downloading Reports ......... 174
Parameters for .................. 175–77
Sample Profiles .............. 177–78
Uploading Reports .......... 175
Uploads and Downloads ...... 178
CFXPowellX .......... 165–72
Commands ..................... 172
Generic Transfers .......... 169
CFXPowellX
Command Processing Parameters 169
Download Specifications .... 171–72
Parameters for .................. 166–69
Upload Specifications .... 170–71
Control Block .............. 150–57
Logical Record .............. 149
Overview ..................... 5
Parameters .............. 43
Sample ...................... 150
Security Exit .......... 185–90
Control Block .......... 186–87
Sample ...................... 187–90
User ID .................... 4. see also PD ID
Add New ....................... 26
Delete ......................... 27
Maintenance .................. 26
Record ...................... 25
Update ....................... 27
User Profile .............. 76. see Profile
Access Ranges and Limits .... 68
Output Format ............. 69
User-ID .................... 24

V
Variable Parameters ........ 43, 76
for User Exits .................. 43
Variable Table .............. see Tables

W
WKS Format ................... 38
Write Limit ................... 39

X
xBASE ....................... 202
xBASE Format .......... 38, 54, 55, 56
Downloads .................... 54