



MultiCICS Administration Guide

Release 3.6.0

UNICOM
SYSTEMS, INC.

Document Number PIEMC0400-01

First Edition printed August 2007

©2007 - UNICOM Systems, Incorporated
All Rights Reserved

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, without written permission from UNICOM Systems, Inc.

This manual applies to PIE/CICS release 3.6.0 and to all subsequent releases of the product until otherwise indicated by new editions or updates to this publication.

All product names mentioned are trademarks of their respective companies.

UNICOM Systems, Inc.
UNICOM Plaza - Suite 310
15535 San Fernando Mission Blvd.
Mission Hills, California 91345

Contents

About This Manual.....	vii
Audience	vii
How This Manual is Organized	viii
Recommended Reading	ix
Common Manuals	ix
Optional Manuals.....	x
Syntax Conventions	xi
Chapter 1 MultiCICS Introduction	1
Benefits to End Users	1
Making CICS User Friendly	2
Sessions Menu	2
Single Signon	2
Eliminating the Blank CICS Screen.....	3
Transferring Screen Data Between Sessions.....	3
Scripts.....	3
Flexible System.....	4
System or User Control.....	4
Chapter 2 MultiCICS Customization	5
Step 1: Planning	6
Step 2: Accessing PIE as an Administrator.....	7
Step 3: Set Up Customization Options.....	8
Step 4: Multiple System Support.....	10
Step 5: Accessing Other VTAM Applications.....	11
Step 6: Compatibility With Other Software Products	12
Step 7: Edit the User Directory.....	13
Step 8: Edit the Terminal Directory.....	15
Step 9: Set Up Profiles.....	16
Step 10: Set Up the Logon Director	22
Chapter 3 Session Execution String	23
PIE Commands.....	23
CMD and TRAN.....	24
MENU.....	24
BROWSE and EDIT	24
SYSTEM (CICS blank screen)	24
PASS, TSWITCH, and ACCESS	24
PROMPT	24
AutoEnd	25

Command Stacking	27
Executing Transactions and Commands	28
Executing PIE Application Commands	28
Executing Scripts and PIE/REXX Programs	29
Executing Menus	30
Accessing Text Files: Help, News, and Bulletin Boards	31
Help and News	31
Executing an Application in Another VTAM APPLID	32
Special Features of MultiCICS	33
Running a Native CICS Session	33
Using a Dynamic Menu as a Sessions menu	33
NetMizer Optimization	34
Chapter 4 Case studies	35
Case Study 1: System-Wide Implementation	35
Case Study 2: VTAM Applications in Separate Sessions	37
Case Study 3: User Groups	40
Chapter 5 Production and Maintenance	41
Performance Considerations	41
Making PIE/CICS the Good Morning Transaction	41
Ongoing Administration	42
Chapter 6 MultiCICS Basics	43
Sessions Menu	44
Sessions Menu Commands	45
Passing Session Arguments from the Sessions Menu	46
Types of Sessions	47
Attributes of an Application Session	47
Defining Sessions	47
Defining and Accessing Sessions: Exercises	49
Switching Between Sessions	51
What Happens During Session Switching?	51
Sessions menu	51
Escape String	52
Using PF Keys: The SWITCH Command	53
Ending Sessions	54
Closing and Cancelling Sessions	55
Exercises	56
Appendix A Customer Support	57
Contacting Customer Service	57
Troubleshooting Suggestions	58
Describing the Problem	59

Appendix B System Tables	61
Appendix C Exits	63
Appendix D Application Program Interfaces	65
Appendix E Limiting Transactions to One Session	67
Index	69





About This Manual

MultiCICS is an optional PIE/CICS component that provides access up to 99 simultaneous applications running in separate sessions. Using NetGate and MultiCICS together, users can switch between sessions with simple commands, menu options, or a designated PF key. This manual includes information to customize MultiCICS and use it in a typical CICS production environment.

Audience

This book is intended for system administrators responsible for maintaining PIE/CICS at their site. Readers are expected to understand CICS and MVS concepts.

How This Manual is Organized

This manual consists of six chapters and five appendixes. Listed below are the titles and a brief description of the contents of each chapter and appendix.

- Chapter 1 MultiCICS Introduction
Describes NetGate concepts and how it can be used to provide access to multiple VTAM applications
- Chapter 2 MultiCICS Customization
Describes common customization procedures to prepare MultiCICS for use at most sites.
- Chapter 3 Session Execution String
Describes PIE/CICS commands that are used to manage multiple sessions.
- Chapter 4 Case Studies
Provides a series of practical examples that show how to implement MultiCICS.
- Chapter 5 Production and Maintenance
Lists periodic maintenance procedures to adapt MultiCICS to changes in your site's CICS system.
- Chapter 6 MultiCICS Basics
Describes basic operational procedures to start, stop, and navigate between sessions with PIE commands.
- Appendix A Customer Service
Describes procedures to report problems with PIE/CICS to UNICOM Systems, Inc. Customer Service.
- Appendix B System Tables
Describes system tables that are used by MultiCICS.
- Appendix C Exits
Describes PIE/CICS exits that are used by MultiCICS
- Appendix D Application Programming Interfaces
Describes application programming interfaces that are used by MultiCICS.
- Appendix E Limiting Transactions to One Session
Includes a procedure to restrict transactions to execute in a single session.

Recommended Reading

The title and a brief description of all PIE/CICS manuals are shown in the following lists. Some manuals provide common information that applies to both the common and optional components of PIE/CICS. Other manuals pertain only to optional PIE/CICS components. These manuals need to be read only if optional components are part of the PIE/CICS system installed at your site.

Common Manuals

These manuals provide common information that applies to both the shared and optional components of the PIE/CICS family.

- *PIE/CICS Installation Guide*
Includes a series of procedures to install PIE/CICS.
- *PIE/CICS Release Notes*
Describes new features or enhancements to PIE/CICS introduced with Release 3.6.0
- *PIE/CICS Command Reference*
Describes PIE/CICS Application and Environment commands.
- *PIE/CICS Customization Guide*
Describes common procedures to adapt PIE/CICS to your site's environment.
- *PIE/CICS Operation and Administration Guide*
Describes common features or facilities that are available to all PIE/CICS products. Performance tuning techniques and implementing security also are described.
- *REXX for PIE/CICS User Guide*
Describes how to write, compile, and execute SAA-compliant REXX programs that operate within a PIE/CICS external environment.
- *PIE/CICS Custom Menus Administration Guide*
Describes how to create custom MultiCICS and Dynamic Menu screens that provide alternate language support.

Optional Manuals

These manuals describe optional PIE/CICS components.

- *PIE/CICS MultiCICS Administration Guide*
Provides customization procedures and usage information to support multiple PIE/CICS sessions with MultiCICS.
- *PIE/CICS Dynamic Menus Administration Guide*
Describes how to create custom PIE/CICS menus that provide extended security and enhanced transaction processing.
- *PIE/CICS NetGate Administration Guide*
Explains how to access multiple VTAM applications through a PIE/CICS session with NetGate.
- *PIE/CICS NetMizer Administration Guide*
Describes how to use NetMizer to optimize 3270-based data streams across a network.
- *PIE/CICS Availability Plus Administration Guide*
Explains how to use Availability Plus to distribute and balance work across multiple CICS regions.
- *PIE/CICS NonStop CICS Administration Guide*
Describes how to use NonStop CICS to route work across CICS regions to balance the workload and minimize down time in the event of a region failure.

Syntax Conventions

A syntax diagram is included with each PIE/CICS command described in this manual. A syntax diagram shows the possible parameters, values, and variables associated with a command.

Syntax diagrams adhere to common conventions. The physical appearance of a diagram's elements indicates whether a command parameter, variable, or other values are required, optional, or included by default.

- An underlined parameter is the default assigned to the command.
- Command names are presented in MIXed case. The uppercase portion of a command name is the requisite abbreviated form. Lowercase letters represent the optional remainder of the command name that need not be specified to execute the command.
- An *italicized lowercase* parameter represents a value assigned by the user.
- A vertical bar (|) separates two or more mutually exclusive parameter values. Only one value can be specified for each parameter.
- Parameters enclosed within brackets [] are optional. Only one value can be specified to a parameter.
- Parameters values enclosed within braces { } are required. If unspecified, the parameter default is assigned to the command.
- Monospace type indicates a screen field or an example of a PIE/CICS command entered on the screen.



Chapter 1 MultiCICS Introduction

Most end-users work with several different transactions during the course of a business day. Often times, users who work with customers and vendors may use a series of transactions during the same call.

With CICS alone, users must end their current transaction before they can access another. A significant amount of time and CPU resources are spent just starting and ending transactions. For example, a clerk may have to complete the following sequence of tasks to process a parts order:

- Open a customer database to access customer information
- Close the customer database
- Start an inventory database to look up a part number
- End the inventory database
- Start an order entry transaction to process the order

MultiCICS simplifies the work required to process the parts order by giving end-users near simultaneous access to different applications with a single logon. Each application runs in a separate MultiCICS session. Sessions can also run applications in different VTAM APPLIDs, such as stand-alone CICS regions, TSO, and IMS. You can even run the same application in multiple sessions to provide quicker access to different application screens.

Users switch between sessions with simple commands, pop-up menus, or PF keys. In the example of the order entry clerk processing a parts order, the work is simplified to:

- Switching to the customer database session to access customer information
- Switching to the inventory database to look up the part number
- Switching to the order entry transaction to enter the order

If the applications are running in separate sessions, switching between sessions takes about the same amount of time as it takes to refresh an application screen.

Benefits to End Users

MultiCICS is resource efficient.

- Up to 99 sessions are available to each user with a single logon. MultiCICS uses a single TCTTE. No extra address space or virtual terminal is used.
- Only one session is active at a time.
- Users can switch sessions with a single keystroke.
- Screen images are maintained in memory, eliminating the need for a read buffer request when users switch between sessions. This avoids a great deal of network traffic, which can slow response time.

Because MultiCICS is resource efficient, you can make it available to your entire end-user community. You don't have to reserve it for data processing personnel. MultiCICS benefits your entire company, not just a small group.

Making CICS User Friendly

MultiCICS makes it easy for you to simplify user tasks and make user applications more accessible. Your users can accomplish their work with less time and effort.

The following features simplify and automate end-user processing:

- A Sessions menu provides easy access to commonly used applications and systems.
- A single signon provides access to multiple systems.
- Eliminating the CICS blank screen reduces user confusion and help desk calls.
- Cut and paste operations make it easy to transfer data from one session screen to another.
- Scripts automate repetitive user tasks.

Sessions Menu

The Sessions menu shows all sessions available to users. To start an application, users choose the corresponding option from the Sessions menu.

```

PIE Sessions Menu ----- (C) TSC, Inc 1987
Command ===>
..#.. Name.... Title..... Esc Str: ==
_ 1 ACTION      Action Items      User: SD080BB
_ 2 TPRODUCT    Trials by Product Term: 0003
_ 3 TNAME        Trials by Customer Name   H Node: PIE10003
_ 4 CPRODUCT    Customers by Product     Sys : TSCDB
_ 5 CNAME        Customers by Customer Name H CPU : 168A
_ 6 DATABASE     Data Base/Email         H Ver : 02.03.00
                               Time: 21:24:38
                               Date: 06/14/96
                               Sun Oct 14
                               Exit: PA1

PF: 1 Help 2 Open 3 End 4 Close 5 Switch 6 PIE 10 View 11 Bulletin 12 Profile

```

The Sessions menu can be customized at the user, group, and system levels. For example, you can set up a default Sessions menu for general users, an Accounts Payable and Receivable menu for your accounting department, and custom menus that are adapted to the needs of individual users.

Single Signon

Users sign on to PIE/CICS using the Logon Director. The Logon Director stores the user's ID, password, profile, and other information. Whenever a subsequent signon is required, the Logon Director can pass the required information automatically and transparently.

Eliminating the Blank CICS Screen

MultiCICS allows you to eliminate the CICS blank screen. As a result, users don't get "lost" and have to call the helpdesk as often. It also means they are not as free to experiment.

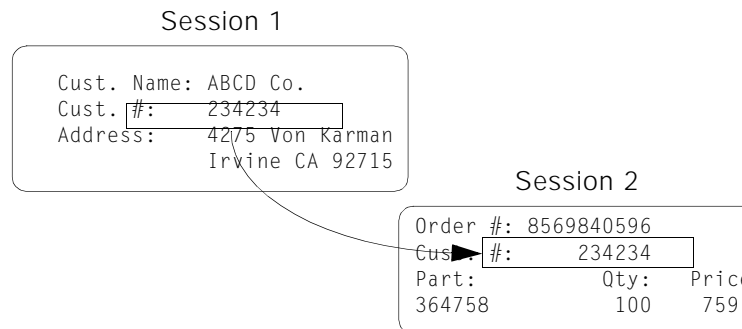
The following features can be implemented to prevent the CICS blank screen from being shown to users:

- AutoEnd
- Predefine all sessions
- Restrict access to the CANCEL, FREE, and SYSTEM commands

Transferring Screen Data Between Sessions

Frequently, users need to transcribe data from one application to another. For example, suppose customer data is already entered in one application, and your user needs to copy it to another application to process an order.

Instead of writing the information down, switching to another session, and re-typing it, users can copy the information from one session screen, switch sessions, and paste the information onto the new session screen.



Scripts

Frequently, users perform the same sequence of steps each time they complete a certain task. For example, let's say the user logs on to a new CICS region, starts a transaction, and progresses through three screens before reaching the destination screen. The process would be quicker and more convenient if it were automated. With PIE/CICS, you can automate it. You can create a PIE/REXX program or a "play-back" script.

PIE/REXX is a programming language available with PIE/CICS that is compliant with IBM's SAA REXX. In addition to traditional REXX commands, PIE/REXX includes a set of special environment commands that allow you to view and manipulate screens and perform other CICS functions. PIE/REXX can interact with applications, make decisions, retrieve, and update information from other sources, even create simple composite applications.

With play-back scripts, you simply record the keystrokes used to complete the task. You can use PIE/CICS variables for variable data; for example, user IDs and passwords. The script records keystrokes along with variables. When you execute the script, it executes your keystrokes and substitutes current values for the variables.

Using either PIE/CICS facility, you can create programs and scripts that automatically execute just about any end-user procedure. For example, you can create a script to:

- Navigate users through a series of screens to get to the function they need
- Supply variable information to a function automatically
- Sign on users automatically to an application that requires security validation

Flexible System

MultiCICS is flexible. With the flexibility built right in to the product, some customized user-exits, and a little help from us, you can customize it to fit almost any set of requirements. For example, you can

- Predefine user sessions or let your users define their own sessions
- Create custom switch sequences and PF keys for different users and groups
- Handle an application differently in different sessions—for example, change variable data or run it in different AORs
- Allow users to run the same application in many sessions or limit certain transactions or groups of transactions to a single session
- Make all commands available to all users or make some commands off-limits to certain users
- Make PIE/CICS features, such as the Sessions menu and pop-up menus, fully available to users or make PIE/CICS completely transparent

System or User Control

MultiCICS system administrators ultimately control the MultiCICS environment. As a system administrator, you determine who can sign on and what access users have to system resources. You can designate group administrators to maintain PIE resources for their groups and authorize individual users to maintain private resources for themselves. Or, you can retain complete control over all PIE resources as the system administrator.

Those in control of PIE resources determine how many sessions users can have open at once, what they can run in their sessions, and whether they can open sessions for themselves. They can also write and implement REXX programs and scripts.

PIE/CICS flexibility provides both the centralized control of a mainframe, with its advanced security and efficiency, and decentralized control to allow users to modify their working environment to suit their individual needs.

Chapter 2 MultiCICS Customization

This chapter describes MultiCICS customization procedures. MultiCICS should be fully operational after completing these procedures.

This chapter presents customization from the perspective of fully implementing MultiCICS as part of your PIE/CICS system. However, you should implement MultiCICS in a series of planned stages. You should begin by implementing only a few users or user groups and a single, simple profile. Test your system to verify that MultiCICS has been implemented correctly. After you have become familiar with MultiCICS, add more users and sessions. After the first several stages, most of the remaining work pertains to changing the User Directory and profiles. Continue this process until the system is fully implemented.

PIE/CICS and its optional products use many shared features and facilities. The *PIE/CICS Customization Guide* describes the procedures to prepare these shared features. The procedures in this chapter make frequent references to the *Customization Guide*. You will need to refer to the *Customization Guide* for information to complete some of the procedures in this chapter.

Step 1: Planning

This step lists a series of questions that you should answer as part of the planning phase to customize your MultiCICS system. At this point you don't need detailed answers to these questions. Just get an idea of what you will need to implement later. As you consider session configurations, you may find that different users differ widely in the transactions they need. If that is the case, it is probably best to set up user groups, and you may want to change your answer to question number 4.

As you plan, look at the sample scenarios in ["Chapter 4 Case studies" on page 35](#). These case studies present several typical scenarios with accompanying procedures to customize MultiCICS. Use the case studies as examples of the overall process to customize MultiCICS.

1. Do you want a system that is highly specialized for different groups of users or generalized for system-wide use?

Specialized systems are more convenient for the user and help them perform their work faster. Generalized systems are easier to prepare and maintain.

2. How many user groups, if any, will you implement?

You can have different user groups for different departments (payroll), job descriptions (manager), corporate divisions (Western region), and so forth.

3. Develop a consistent naming convention for your user groups.

4. If you are implementing user groups, more specific user masks must be entered in the User Directory. Do you want to associate maintenance of this directory to maintenance to your security system?

See [" PUXINIT—Post Sign on Initialization" on page 149](#) of the *PIE/CICS Customization Guide* for details.

5. Plan session configurations for each group of users. Determine what transactions users need and the number of sessions they need to have open at one time.

Are you using Dynamic Menus? You might want to implement a menu in one session and have it run its applications in new sessions.

6. Do users have repetitive tasks that could be automated by a script?

7. Do you want users to have private object authority? That is, do you want users to be able to save changes they make to their run-time profile? Do you want them to be able to create scripts and other private objects?

8. Do you want PIE/CICS to start automatically as the good morning transaction or do you want users to request PIE/CICS to have access to MultiCICS?

Having PIE/CICS come up as the good morning transaction is the preferred option in most circumstances. However, you may not want it during implementation and testing.

9. How will you set up the Logon Director?

The Logon Director has several implementation options. They are discussed in ["Chapter 2 Logon Director" on page 3](#) of the *PIE/CICS Customization Guide*. We recommend that you set up the Logon Director as the full panel signon system for CICS and PIE/CICS.

After you have answered these questions, you should have a fairly good idea of the overall plan to implement MultiCICS.

Step 2: Accessing PIE as an Administrator

For many of the customization steps described in this chapter, you need access to PIE/CICS as a system administrator. System administrator authorization is specified in the User Directory. The default directory shipped with PIE/CICS authorizes all users as system administrators.

Most customization can be performed online, from the PIE/CICS system administrator menu shown in the following figure. To access this menu, perform one of the following.

- If your PIE/CICS system is new, access PIE/CICS by entering the PIE transaction. Sign on at the Logon Director screen. You will be presented with the PIE/CICS main menu. Select option 1, the System Administrator menu.
- If your PIE/CICS system has already been customized for MultiCICS or another component, enter PEXE MENU PIEMMSTR at a CICS blank screen or MENU PIEMMSTR at the Sessions menu command line.
- Select option 1, the System Administrator menu, from the main menu.

☞ Throughout this manual we assume you are working on a newly installed system with the default options shipped with the product. If your site has PIE/CICS currently installed, your options and directories may be somewhat different than the examples shown in this chapter.

```

System Administration Facility -----
Option ==>                               Scroll ==> CUR
..#.. Name.... Title..... Arguments.....
      User=USR1 Terminal=PIE10015
_  1 NEWS      - Edit System News on Tue 09/10/96 8:02:13
_  2 TERMINAL - Update Generic Terminal Directory
_  3 USER      - Update Generic User Directory

_  4 LOGON     - List/Edit Logon Panels
_  5 SCRIPT    - List/Execute/Browse Script files          *
_  6 TEXT      - List/Edit Text Files;          GROUPID ==> *
_  7 MENU      - List/Edit Dialog Manager Menus          *
_  8 PROFILE   - List/Edit Profiles                      *

_  9 LICENSE   - Specify License Information for PIE
_ 10 OPTIONS   - Customize PIE System Options
_ 11 LOAD      - Upload/Download PIE objects

_  C CICS      - Native CICSMZW1 Environment
_  X EXIT      - Terminate This Menu
      **END**

F1=Help      F2=SElect  F3=END      F4=RETURN   F7=UP       F8=D0wn
    
```



Step 3: Set Up Customization Options

Many system-wide defaults are specified with both PIE/CICS Customization Options screens. This step explains how to specify the customization options that pertain to MultiCICS. For more information about other Customization Options fields, refer to [“Chapter 5 Customization Options” on page 65](#) of the *PIE/CICS Customization Guide*.

1. Access the Customization Options utility. Select option 10, OPTIONS, from the System Administrator menu.

The following figure shows an example of the first Customization Options screen.

```

PIE Customization Options (page 1 of 2)----- (C) 1995 TSC, Inc
Command ===>

Maximum logon attempt count . . . . . 8          Version 03.03.00
Avg No. of Concurrent Users . . . . . 1          Date 06/19/99
TSWITCH Auto-Signon Timeout . . . . . 1 (Minutes) Jul 96.171
Avg No. of Sessions per User . . . . . 1          APPLID CICSPROD
Temp storage name substitution . . . NO
MRO/ISC Support . . . . . FUL
Temp Storage on Auxiliary . . . . . YES
Name of Logon Director Help . . . . . HELPHELP
Transient TCTTE's . . . . . YES
AOR AE/ATI support-1 byte field . . . 4 (decimal offset in the TCT user area)
Create Tctte Extension . . . . . YES
or Use 4 byte field at . . . . . (decimal offset in the TCT user area)
Temp Storage Queue Prefixes . . . . . ZZZ
Load Balancer 16 byte field at . . . . 5 (decimal offset in the TCT user area)
Automatic PIE GMM Tran . . . . . YES
Message Destination Id . . . . . PIEL
Auto-start Network Monitor . . . . . YES

INSTRUCTIONS: Enter Down command for more options

```

2. Edit the following screen fields.

Avg No. of Sessions per User

Specifying this value correctly helps minimize the resources allocated for PIE/CICS and improves performance.

Temp storage name substitution

Indicate whether you want PIE/CICS to substitute unique temporary storage names for each session. YES is required to:

- run the same application in more than one session
- port a sign on from one terminal to another, keeping sessions and work intact

MRO/ISC Support

There are four valid values for this field: YES, FS, FUL, and NO. Since there is some overhead associated with each type of support, specify only the support you need.

If you are not running MRO/ISC, code NO.

If you are running MRO/ISC, consider each of the following conditions. These conditions may suggest more than one value for the field; for any combination of YES, FS, or FUL, code FUL. If none of the conditions apply, code NO.

- To use AutoEnd with remote transactions that do ATI (EXEC CICS) STARTs back to the terminal, specify FUL.
- To use the PIEEXEC API or the Retrieve User Data API in an AOR, specify YES or FUL.
- To support TSQ substitution in an AOR, see the table below. (PIE/CICS will use either transaction routing or function shipping. The method is determined by the application requesting the TSQ.)

Transaction Routing	Function Shipping	Code
YES	NO	YES
NO	YES	FS
YES	YES	FUL

- For pseudo termid support, specify YES. Pseudo termid support gives each session its own terminal ID in the CICS control blocks. When you use pseudo termid support, you do not need temporary storage queue substitution.

3. Press PF8 to scroll forward to the second Customization Options screen.

4. Edit the following fields of the second Customization Options screen:

Limit Transactions to One Session

You may set this to YES or NO. It must be YES if you want to limit certain transactions to execute in only one session at a time. You must add those transactions to a table. By default, no transactions are restricted to a single session. See ["Appendix E Limiting Transactions to One Session" on page 67](#) for more information.)

Generic matching on Menus/Sessions

Specify YES if you want users to be able to identify a session with a shortened version of its name.

Action on max sessions

Specify whether you want users to go to the Sessions menu or the Switch pop-up menu if they try to define a new session but have no more sessions available.

Action on session close

Specify where you want users to be placed when they end a session. Specify LAS to send them to the last application session they were in or the Sessions menu, whichever was accessed last. Specify BSM (bypass Sessions menu) to send them to application sessions only, never the Sessions menu. LAS is the most common choice.

5. Press PF5 to save your changes after you have finished making changes to the screen fields.

☞ PF3 does not save your changes. Be sure to press PF5 before exiting.

6. Press PF3 to exit the screen.

Step 4: Multiple System Support

If you are running MRO/ISC or multiple stand-alone systems, you must specify how the PIE Repository is shared between regions. Refer to the *PIE/CICS Customization Guide*, "[Chapter 8 Multiple CICS Regions](#)" on page 101 for a description of the procedures to share the Repository between regions.

Step 5: Accessing Other VTAM Applications

You can access other VTAM applications with MultiCICS using three PIE/CICS commands:

- PASS
- TSWITCH
- ACCESS (NetGate users only)

PASS and TSWITCH use CLSDST PASS to the target region. NetGate uses a virtual terminal, like a VTAM session manager.

All three commands require some CICS and VTAM customization before you can use them. See "[Customizing Transfer Commands](#)" on page 113 of the *Customization Guide*, for procedures.

Step 6: Compatibility With Other Software Products

Some software products require customization, either to themselves or PIE/CICS before they can run under PIE/CICS. See ["Appendix C Software Compatibility" on page 169](#) of the *Customization Guide* for requirements.

Step 7: Edit the User Directory

The User Directory determines which users can sign on to PIE/CICS. It also determines PIE/CICS characteristics for those users, such as their profiles and how many sessions they can have open at once.

It is important to edit the User Directory as soon as possible after installing PIE/CICS. The directory shipped authorizes all users as system administrators. You should limit that authority to appropriate users only.

☞ Be careful making changes to the User Directory. If you delete all valid entries, no one will be able to access PIE/CICS—not even a system administrator—or if you delete all administrator entries, no one will be able to change the User Directory. In either case, you must reload the User Directory from a backup Repository.

1. Select option 3, USER, from the System Administrator menu to access the User Directory.

The following screen is an example of the PIE Generic Users Directory.

```
PIE Generic Users Directory ----- (C) TSC, Inc 1995
Command ==>                               Scroll ==> DATA
... Userid..   Terminal   Password   Groupid.   MenuName   Profile.   Updated.
___ (MODEL)   (MODEL)   (MODEL)
___ *         *         *         *         PIEMMAIN   USER      03/17/99
___ *         *         *         *         PIEMADM    ACCOUNT   03/17/99
___ *         *         *         *         PIEMMAIN   USER      01/27/99
**END**
```

2. Tab to the MODEL entry and press PF2 to see the Detail screen.

The MODEL entry specifies the defaults for other entries that have blank fields. For example, if you have a user entry with a blank Groupid field, PIE/CICS uses the Groupid value from the MODEL entry for this user.

Change the fields as you like. See ["User Directory" on page 52](#) of the *PIE/CICS Customization Guide* for information on this screen.

One of the fields requests the name of a profile. You will create these later in the customization procedure. At this point, you can create profile names and write them down for future reference or you can return to this screen and fill in the Profile field after you have created the profiles.

The Help desk ID field is also worth special notice. This field accepts a user ID mask. Anyone with a matching ID can view the screen of a user logged on with an entry using that value. Since the field in the MODEL entry often affects a large number of users, consider carefully what you enter in this field. If you leave the field blank, only system administrators and the group administrators for this group will be able authorized. See ["Restricting Access to VIEW" on page 117](#) of the *Operations and Administration Guide*, for information on other ways to grant or limit view authority.

The Maximum Sessions, Security Classes, and Allow Session Cancel fields all apply particularly to MultiCICS. The Maximum Sessions field determines the maximum number of sessions this user can have defined at one time. The Security Classes field determines which PIE Environment command the user is authorized to execute. Leave it blank to authorize all commands or see the *PIE/CICS Command Reference* for security classes. The Allow Session Cancel field determines whether users can end sessions that contain active applications. That includes issuing the

CANCEL, CLOSE, and EXIT SIGNOFF and LOGOFF commands. If `Allow Session Cancel` is NO, the user will be requested to end the application normally before closing the session.

When you are finished, press PF3 to exit the detail screen. You can enter SAVE on the command line to save your changes at this point, if you like.

3. If your PIE/CICS system is new, you must limit system administrator authority.

Tab down to the first * entry. (This is the system administrator entry.) Type your user ID in the `User id` field.

With the cursor still in this entry, press PF2 to access the detail screen. Change fields as you like. However, the `System Admin` field must specify Y.

When you are finished, press PF3 to exit the detail screen.

If you want to authorize more system administrators, place the cursor in the command field of the entry you just modified, type R, and press ENTER. Now you have two identical entries. Modify this entry the same way you modified the first.

Continue with this procedure until you have created all the system administrator entries you need.

4. Modify your user entries.

The remaining * entry (there should be only one now) authorizes all users to access PIE/CICS as standard users, without administrative authority.

Place the cursor on this entry and press PF2 to see the Detail screen. Change fields as you like. We recommend that you leave the `User id` field on this entry an asterisk, so all users have access to PIE/CICS. When you are finished, press PF3 to exit the detail screen.

5. If you want to set up user groups, place the cursor in the command field of the * entry. Type R and press ENTER.

Now you have two identical user entries. The `first` entry is the one to change. (The * entry, being the most general, should always be last.) Enter a user ID mask in the `User id` field, to indicate the users in this group. Press PF2 to access the detail screen and make changes as before. When you are finished, press PF3 to return to the primary screen.

Continue creating new entries until you have all the user groups you need.

☞ Often the only difference between one user group and another is the profile used. If that is the case with your user groups, you may not need to enter each user group in the User Directory. Instead, set up the PUXINIT exit to specify user profiles dynamically when the user logs on. For this to work, you must have user groups set up in your security system. See the *PIE/CICS Customization Guide*, "[PUXINIT—Post Sign on Initialization](#)" on page 149, for more information.

6. When you have finished making changes, press PF3 to save your changes and exit the screen.

Step 8: Edit the Terminal Directory

The Terminal Directory determines which terminals can sign on to PIE/CICS. It also determines PIE/CICS characteristics for those terminals, such as the logon panel displayed and whether users can log off CICS from this terminal.

The default values set in the Terminal Directory are often acceptable without any further required changes. The default values authorize all terminals to sign on to PIE/CICS with the default logon panel.

- To edit the Terminal Directory, access the Terminal Directory Utility. Select option 2, TERMINAL, from the System Administrator menu.**

You will receive the following screen:

```
PIE Generic Terminal Directory ----- (C) TSC, Inc 1995
Command ==>                               Scroll ==> DATA
... Terminal Userid.. Password Logon  Logoff PIE Dial Opt SLM TRM RST Updated.
___ (MODEL)                PIELMAIN USE  YES NO  FUL          03/37/99
___ *                      PIELMAIN      YES NO  FUL NO  FUL NO  01/27/99
**END**
```

- Edit lines and fields as you like. There are no specific requirements for MultiCICS.**

Logon screens are assigned from the Terminal Directory. If you want different terminals to use different logon screens, create separate entries for these terminal groups and specify a logon screen name in the Logon field. You will create and edit logon screens later in the customization procedure. You might want to make a note of the screen names you used here.

If you edit these entries, be sure to see "[Directory Entries](#)" on page 45 of the *PIE/CICS Customization Guide* for information about entry order. Also, refer to "[Terminal Directory](#)" on page 46, for information about entry fields.

☞ Be careful making any changes to the Terminal Directory. If you delete all valid entries, no one will be able to access PIE/CICS—not even a system administrator. You will have to reload the Terminal Directory from a backup Repository.

- When you have finished making changes, press PF3 to save your changes and exit the screen.**

Step 9: Set Up Profiles

In this step you will set up your user profiles. Profiles customize the user's working environment. They define:

- Users' initial session configuration
- Default application for new sessions
- what happens when users first log on
- Final PIE/CICS command
- Default PF keys

As you perform the procedures below, see ["Chapter 3 Profiles" on page 29](#) of the *PIE/CICS Customization Guide* for more information on profiles, the profile utility, and profile fields.

1. Access the Profile Utility. Select option 8, PROFILE, from the System Administrator menu.

The following figure shows an example of the Profile List.

```

PIE Utility list of PROFILES -----(C) TSC, Inc 1995
Command===>                Lines  1 to 18 of 33

Sel. Group... Name.... Changed..... By..... Size. Title.....
  *  *  *      *  *  *
__ SYSTEM ACCOUNT 95/04/18 11:56 DEMO    8 System Administrator profi
__ SYSTEM DEMO   91/11/14 19:25 SD080BB  6 Demo user profile from Pri
**END**

F1=Help F2=Add F3=End F4=RETu F6=REFr F7=Up F8=Down F10=Left F11=Right

```

2. We recommend that you begin by creating one profile as a model for others. The Profile Utility does not have a model entry like the Terminal and User Directories. However, you can create a "model" profile and copy it to create other profiles.

If your system new, it is best to keep your profile simple. When you have tested it, you can come back and add improvements.

To create a new profile with defaults only press PF2. You will receive the Profile Utility's main screen.

```

PIE Profile (SYSTEM.TECH) ----- (C) TSC, Inc 1987
Option ==>

      1 Terminal - Specify Terminal Characteristics
      2 Defaults - Specify defaults for PIE (reserved)
      3 Keys     - Specify Global PIE keys
      4 Sessions - Specify Sessions Configuration

Update History:
Name:  SYSADMN
Term:  PIE10002
Date:  09/03/96
Time:  9:34:50

PIE profile group ==> SYSTEM
PIE profile name  ==> TECH
PIE profile title ==> System Maintenance

Default Application ==> menu
Initial PIE command ==> sessions
Final PIE command  ==> exit

Administrator ==> SYSADMN      Terminal ==> *
Auditor        ==> *           Terminal ==> *
Generic forms, ?, % and * characters are accepted above.

PF: 1 HELp  3 END  4 RETUrN  5 SAVe  6 RETRIeve  9 DELete

```

3. If this profile is for a user group, type a name in the PIE profile group field.

Type the profile name in the PIE profile name field. In [“Step 7: Edit the User Directory,” on page 13](#), you may have established profile names by assigning them to user entries. If so, be sure to save your profiles with the names you used there.

Type a description of the profile in the PIE profile title field. This description will be displayed with the profile name in the Profile List screen.

4. Specify values for the following fields.

Default Application

The default application will be executed when a session is opened but no application is specified to run in it. Typically, a menu is specified in this field. If you leave it blank, the SYSTEM command will be executed and the user will get the CICS blank screen.

☞ This field expects a PIE Application command. If you want to execute a PIE Environment command, prefix it with EC.

The Default Application will be used whenever you create a session without execution text. If the Default Application field is blank then the SYSTEM command will be executed, causing that session to enter into native CICS (blank screen).

Initial PIE command

The initial PIE command is executed when the user signs on to PIE/CICS. If this field is blank or is coded incorrectly, it defaults to OPEN 1. If you want the user to see the Sessions menu, type SESSIONS. You can specify the OPEN command to open a session when the user signs on. For example, if you want the user to see the inventory application, you could type OPEN 1 here and later define session 1 to execute the inventory application.

Final PIE command

The final PIE command is executed when users end their last sessions. Typically, customers specify EXIT LOGOFF, to log users off PIE and CICS.

☞ We recommend that you leave the Administrator and Auditor fields with an asterisk (*).

5. Press PF5 at any time to save your changes.

6. Enter 1 in the command line to access the Terminal screen.

```
PIE Profile Terminal Characteristics----- (C) TSC, Inc 1995
Command ==>

Pie Escape String ==> ==          (1-5 char. start at 2nd position)
Command Delimiters ==> ;          (Characters for command stacking)
Titles display ==> YES            (Yes - display titles)
Message ID ==> NO                 (Yes - display message ID)
SYSTEM END Key ==> PF3            (PF01-PF24, PA01-PA03, CLEAR)
SYSTEM END Key ==> PF15           (PF01-PF24, PA01-PA03, CLEAR)
Scroll Default ==> CURSOR         (Page, Data, Half, Cursor are valid)

INSTRUCTIONS:

    Press ENTER key to update terminal options
    Enter END or RETURN command to update terminal options and exit

PF: 1 HELp  3 END  4 RETUrN  5 SAVe
```

7. Change fields as you like. When you have finished, press PF3 to return to the Profile Utility main screen.



8. Enter 3 in the command line to access the Keys screen.

```

PIE Profile Global PIE Keys ----- (C) TSC, Inc 1995
Command ==>                               Keys ==> ON
PIE Key.. Command.....
PF 1 ==>
PF 2 ==>
PF 3 ==>
PF 4 ==>
PF 5 ==>
PF 6 ==>
PF 7 ==>
PF 8 ==>
PF 9 ==>
PF10 ==>
PF11 ==>
PF12 ==>

PA 1 ==> exit
PA 2 ==>

INSTRUCTIONS:
  Enter UP or DOWN Command to view alternate set of keys
  Enter END or RETURN Command to process changes and exit

PF: 1 Help  3 End  4 RETurn  5 SAVe  7 UP  8 DOWN

```

9. Specify PIE global keys. See the *PIE/CICS Command Reference for PIE/CICS commands*.

The keys you define here will override each session's application keys. You define new keys sparingly. Many customers specify only PF 13-24 as PIE global keys, leaving PF 1-12 for applications. Press PF8 to access the second Keys screen and the higher PF keys.

We suggest that you define a key to execute SWITCH, SWITCH NEXT, or SWITCH PREV. When you assign these commands to PF keys, users can press a key to switch from one session to another. Please see "[Switching Between Sessions](#)" on page 51, or the SWITCH command, in the *PIE/CICS Command Reference*, for more information on this important command.

Do not use PA keys for switching. Data that is typed on the session screen but not entered with an AID key, will be lost if you switch with a PA key. This is a 3270 hardware limitation. When a PA1, PA2, or PA3 key is pressed, new screen data is not sent to CICS; so MultiCICS is unaware of this data and cannot save it.

You can set a key to toggle PIE global keys on and off. (The toggle key remains on when other keys are off.) To do so, type TOGGLE key for a PF or PA key, where key is the key assigned.

When you have finished, press PF3 to return to the Profile Utility main screen.

10. Enter 4 in the command line to access the Sessions screen.

```

PIE Profile Sessions Configuration ----- (C) TSC, Inc 1987
Command ==>
      Name....  Title.....  Hide
  1 Name ==> ADMINSTR - System Administration Functions  &ZLname -
Execute ==> menu piemadm (&arg

  2 Name ==> SYSTEM   - System Management Functions      &ZLname -
Execute ==> menu piemsys (&arg

  3 Name ==> CICS     - Native &ZSYSID facilities          &ZLname -
Execute ==> system prompt

  4 Name ==> NOTEPAD  - My private Notepad                          &ZLname -
Execute ==> edit &ZUSER

  5 Name ==> USERS    - List/View active &ZSYSID users      &ZLname -
Execute ==> list users

  6 Name ==> SIGNMSG  - Display My signon messages                          &ZLname -
Execute ==> browse pie.signmsg

PF: 1 Help  3 End  4 RETurn  7 UP  8 DOWN

```

11. Use this screen to pre-define user sessions. Each screen defines six sessions. Use PF7 and PF8 to switch from one screen to another. (You can use PF7 to switch from the first screen to the last screen.)

Assign a short name in the *Name* field. The user can specify this name with the SWITCH command or escape string to switch to the session.

Type a description in the *Title* field. This description will display on the Sessions menu.

The *Hide* field determines whether the session is accessible with the SWITCH NEXT and SWITCH PREVIOUS commands. If you specify Hide=Y, those commands will skip the session.

In the *Execute* field, type a command string to execute when the session opens. You can specify transactions and PIE/CICS commands to execute scripts, menus, utilities, etc., "[Chapter 3 Session Execution String](#)" on page 23 provides detailed information on coding the *Execute* field.

☞ Sessions defined here remain *inactive* until the user switches to them—that is, an application you define here is not executed until the user switches to its corresponding session. Merely defining a session does not put load on your system.

12. When you have finished, press PF3 to return to the Profile Utility main screen.**13. Press PF3 from the main screen to save your changes and exit to the Profile List.**

- 14. If you have more profiles to create, move the cursor to the command field of your new “model” profile. Enter C and supply the new profile name. This will create a copy of the selected profile. Edit it, following the instructions above.**

If this is a new system, we recommend that you create one profile for now. When you have tested the system, you can come back and create the remaining profiles needed.

- 15. When you have finished creating and editing profiles, press PF3 from the Profile List screen to exit the Profile Utility.**
- 16. If you did not specify profile names in the User Directory entries, go back and do so now. See [“Step 9: Set Up Profiles,”](#) on page 16, for directions.**

Step 10: Set Up the Logon Director

MultiCICS has no specific requirements for Logon Director operation. We recommend that users sign on manually at the Logon Director screen (unless they are accessing CICS by way of a VTAM session manager or menu system).

See "[Chapter 2 Logon Director](#)" on [page 3](#) of the *PIE/CICS Customization Guide* for more information about the Logon Director and its options.

You may want to customize the default logon screen or create different logon screens for different terminals.

- 1. Access the Logon Panel utility.**
- 2. Select option 4, LOGON, from the System Administrator menu.**

See "[Customizing the Logon Director Screen](#)" on [page 18](#) of the *PIE/CICS Customization Guide* for instructions on this utility.

☞ If you assigned those screens already in the Terminal Directory, be sure to save your new screens with their assigned names.

Chapter 3 Session Execution String

In the figure shown below, the Sessions Configuration menu of the Profile Utility defines the applications that run in MultiCICS sessions. [“Chapter 2 MultiCICS Customization” on page 5](#) introduced the Profile Sessions screen. This chapter describes how to use the Sessions Configuration menu to define the applications that run in MultiCICS sessions.

```

PIE Profile Sessions Configuration ----- (C) TSC, Inc 1987
Command ==>
      Name....  Title.....  Hide
  1 Name ==> ADMINSTR - System Administration Functions  &ZLname -
Execute ==> menu piemadm (&arg

  2 Name ==> SYSTEM   - System Management Functions      &ZLname -
Execute ==> menu piemsys (&arg

  3 Name ==> CICS     - Native &ZSYSID facilities          &ZLname -
Execute ==> system prompt

  4 Name ==> NOTEPAD  - My private Notepad                          &ZLname -
Execute ==> edit &ZUSER

  5 Name ==> USERS    - List/View active &ZSYSID users      &ZLname -
Execute ==> list users

  6 Name ==> SIGNMSG  - Display My signon messages                    &ZLname -
Execute ==> browse pie.signmsg

PF: 1 Help  3 End  4 RETurn  7 UP  8 DOWN

```

PIE Commands

The Execute field of the Profile Sessions screen specifies the command that executes automatically when a session is initially opened. When a user opens the session, the execution string is passed to PIEEXEC, which executes it. Because PIEEXEC gets control first, it can set up the environment as you would like. For example, it can set up AutoEnd; it can clear the screen before a transaction begins; it can parse out stacked commands; and so forth.

Usually, you begin an execution string with a PIE Application command. A few commands are used frequently, such as CMD, TRAN, PASS, TSWITCH, and ACCESS.

In this section, we will explain the common Application commands and what they are used for. Note that these commands are discussed more extensively in the *PIE/CICS Command Reference*.

☞ TRAN is used by default if you code a CICS transaction in the Execute field without specifying a PIE Application command.

CMD and TRAN

You can use either CMD or TRAN before any CICS transaction. The difference between them is that TRAN clears the screen and writes the transaction string on the screen before the transaction is executed. CMD does not; it simply executes the transaction and relies on the transaction to do the required screen clearing.

How do you choose between them? Most transactions clear the screen before they execute, and there is no reason to do it twice. So, for most transactions, CMD is the preferred choice. However, some older transactions do not clear the screen (CRTE, for example). For these transactions, use TRAN to clear the menu from the screen before invoking the transaction.

MENU

MENU displays a selected PIE/CICS menu; either a dynamic menu or one of the standard menus shipped with the product.

BROWSE and EDIT

Use BROWSE and EDIT to access a text file. For example, use BROWSE to view a help text member.

SYSTEM (CICS blank screen)

Use SYSTEM to give the user access to the CICS blank screen.

PASS, TSWITCH, and ACCESS

Use PASS, TSWITCH, and (NetGate) ACCESS when you want to execute a program in another VTAM application. These commands are discussed in ["Chapter 9 VTAM Applications" on page 109](#) of the *Customization Guide*.

PROMPT

Use PROMPT to send a line of text to the user.

When the user presses an AID key, the text can be sent to CICS. Use this to allow the user to modify the execution string. For example:

```
PROMPT CEMT I TERM(    )
```

will display

```
CEMT I TERM(    )
```

The user can fill in the terminal ID:

```
CEMT I TERM(M4A4)
```

and press ENTER to execute the command.

You can also use PROMPT to send a simple message to the user. To do so, you execute the PROMPT command with Confirm End (see ["AutoEnd" on page 25](#)). For example, you could send a message to the user before you execute a transaction. You code the following:

CE PROMPT Use the I option to inquire, the P option to print.;CMD
ABCD

Before MultiCICS executes the ABCD transaction, it sends the user the message:

Use the I option to inquire, the P option to print.

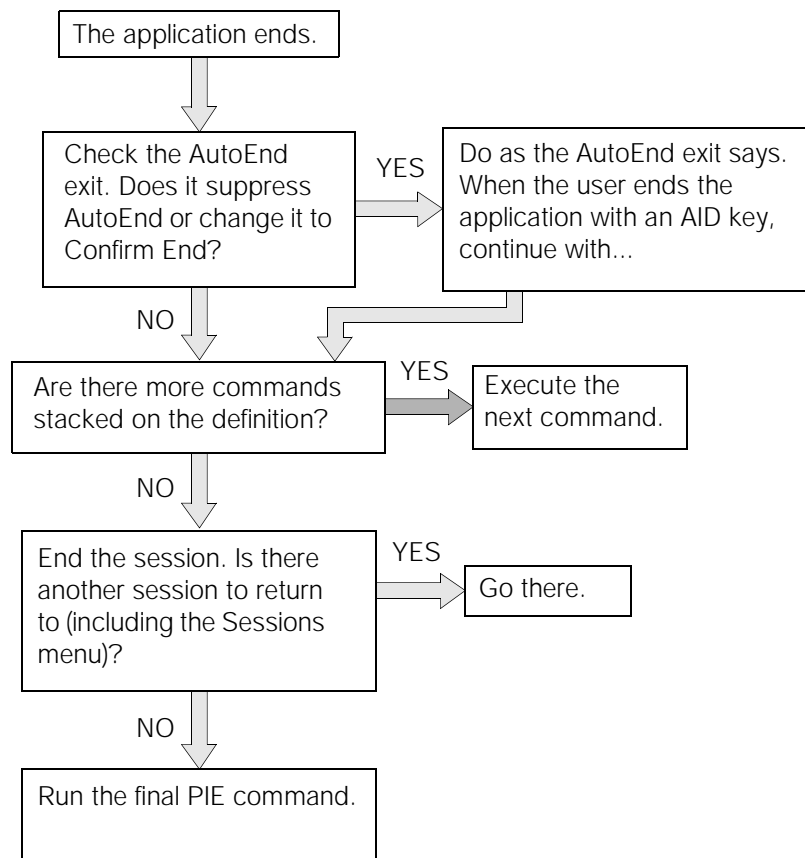
Here the user can make no response, except to press an AID key. Then MultiCICS will execute transaction ABCD.

AutoEnd

AutoEnd returns control back one level when an application ends. For example, suppose you start a session from the Sessions menu, and that session executes your payroll application. When you end the payroll application, AutoEnd returns control to the Sessions menu. With AutoEnd, you don't get the CICS blank screen. You get a session or the Sessions menu instead.

AutoEnd suppresses the final send from the application if that send does not pseudo-converse. So it eliminates blank screens sent by applications when they end. For example, it suppresses the STATUS: SESSION ENDED message sent by CEMT. This streamlines application execution and reduces network traffic.

AutoEnd uses the following logic:



When AutoEnd Works and When It Doesn't

AutoEnd works for conversational, pseudo-conversational, and non-conversational transactions with the following restrictions.

- It works for transactions that pseudo-converse using EXEC CICS RETURN TRANSID(). It does not work for those that pseudo-converse using the next tranid on the screen or in the screen buffer. PIE/CICS will not be able to detect the next tranid and will interrupt the application.
- Some transactions do a final send that is not pseudo-conversational, but is important to the user. AutoEnd is not appropriate for these applications because it will suppress the display. You may use Confirm End in these cases. Confirm End will autoend the application when the user presses an AID key.

AutoEnd is suppressed under the following conditions:

- The transaction is still pseudo-conversing using EXEC CICS RETURN TRANSID().
- System mode is active.
- The terminal is in BMS paging mode.
- An ATI started task is pending.

If AutoEnd is active when an application abends, AutoEnd will be changed to Confirm End so that the user can read the final message sent by the application.

👉 You can change AutoEnd behavior using the AutoEnd exit. See "[PCSSPAEE—AutoEnd Exit](#)" on page 136 of the *PIE/CICS Customization Guide*.

Specifying AutoEnd

AutoEnd is automatic for all PIE/CICS processing. All PIE menus, utilities, and commands (except SYSTEM and PROMPT) run under AutoEnd automatically. For all other applications, you must specify AutoEnd explicitly. Enter the AE state command at the beginning of the menu line execution string. For example, to specify AutoEnd for a CEMT transaction, specify the following:

```
AE CMD CEMT I TAS
```

AE must be placed before CMD, TRAN, or the transaction ID in the execution string. Otherwise, AE would not receive control before the command is issued and would have no way to return control to the menu. You can code other PIE state commands before AE.

As a rule of thumb, it is best to code AutoEnd for all non-PIE/CICS applications. Then, open the session and determine whether AutoEnd works properly with the application. If it does not, edit the execution string and remove AE. Users can return with a system end key. See "[Native CICS: System Mode](#)" on page 27 of the *Operation and Administration Guide*. If you like, you can add those applications to the AutoEnd exit and suppress AutoEnd for them automatically.

Confirm End is set the same way, except that you use the CE command. For example, to specify Confirm End for CEMT, specify the following:

```
CE CMD CEMT I TAS
```

Command Stacking

You can stack commands in a session execution string, so that one command executes after the other. Separate commands with one of the command delimiters defined in the PIE profile being used when the menu is executed. (It is usually best to have a single delimiter common to all profiles.) The default delimiter is a semi-colon (;).

The delimiter is not required if the command has no parameters. For example, AE has no parameters and does not require a delimiter after it. CMD does have parameters, and you must code a delimiter after the CMD string, before the next command.

Let's have some examples. Suppose that you want a user to browse two news files in a row—one for global news, one for local news. You code the session **Execute** field as follows:

```
BROWSE PIEGNEWS;BROWSE PIELNEWS
```

When the user presses PF3 to end the first browse, the second one is executed automatically.

Stacked commands must be controlled by AutoEnd, either automatically or explicitly. The preceding **Execute** field is fine, because the BROWSE command is a PIE command, and it AutoEnds automatically. Suppose you wanted to execute two non-PIE commands, CEDA and CEMT, for example. You would code the **Execute** field as follows:

```
AE CMD CEDA;AE CMD CEMT
```

If the first transaction is non-conversational, the next transaction will be executed immediately. The user does not have to press a key to end the transaction. For example, the PIE command USERDATA is non-conversational. It accepts an argument from the user and places it in the &ZUDATA variable. The following execution string places the argument the user supplies in the &ZUDATA variable and the plays a script called SYSTEM.APACCNT, which requires that variable:

```
USERDATA &ARG;SCRIPT PLAY SYSTEM.APACCNT
```

☞ Execution strings are limited to 66 characters. If your string is longer, see "[Executing Scripts and PIE/REXX Programs](#)" on page 29.

Executing Transactions and Commands

To execute a standard CICS application, you should use AutoEnd and the PIE Application command CMD. With this combination, you avoid useless messages both before and after the application. Users are returned to the menu when the application is ended.

To do so, specify the following in the **Execute** field:

```
AE CMD transaction arguments
```

For example:

```
AE CMD CEMT I TAS
```

```
AE CMD ORDR
```

Executing PIE Application Commands

Executing PIE Application commands from MultiCICS is simple. Simply code the command in the **Execute** field. For example:

```
LIST USERS
```

AutoEnd is automatic for PIE commands, so you need not specify it. The execution string is sent directly to the PIE command processor, so you need not begin the string with an escape string or PEXE.

It does not happen often, but you may need to execute a PIE Environment command in a Session **Execute** field. If you do, prefix the command with **EC**. For example, code the following to change your run-time profile from a line on the Sessions menu:

```
EC PROFILE
```


Executing Scripts and PIE/REXX Programs

Scripts are useful when you open a session. You can create scripts to automatically perform user activities, such as logging on to a TSO system, conditionally navigating through a menu system, or supplying data to screen fields.

REXX programs have some specific benefits. For example, you can code even a simple execution string in a REXX program and execute that program from many different profiles, perhaps from different dynamic menus. Then, if you need to make changes, those changes are effective everywhere that program is executed. You don't have to change each and every menu line. Also execution strings are limited to 66 characters. If you have a longer execution string, you can place it in a REXX program, then execute the REXX program from the Sessions menu.

To execute a PIE/REXX program, simply execute the REXX transaction as described in the PIE/REXX manual. If you want it to be executed in script mode, specify **(SCRIPT** as an argument to the REXX transaction or code **OPTIONS 'SCRIPT'** on the first line of the program. For example:

```
REXX ACCOUNT.PROGRAM (SCRIPT
```

To execute a play-back script, execute the SCRIPT command with the PLAY parameter.

```
SCRIPT PLAY ACCOUNT.PROGRAM
```

REXX is a PIE Application command. SCRIPT is an Environmental command.

See "[Chapter 8 Scripts](#)" on page 93 of the *Operation and Administration Guide* and the *REXX For PIE/CICS User Guide* for information about creating scripts.

Executing Menus

To execute a PIE/CICS menu (either a PIE/CICS supplied menu, such as the PIEADM menu, or a dynamic menu), you simply execute a menu in a menu line using the MENU command. Enter `MENU menuname` in the Execute field.

```
MENU MKTMENU
```

You may use the MAIN parameter after the MENU command, which establishes this as a main menu. PIE/CICS will stop here when the user presses PF4, RETURN.

```
MENU MAIN MKTMENU
```

If you want the user to be able to fast-path to items on this menu, code `(&ARG` after the menu name. (The `&ARG` variable allows the menu to accept the fast-path string and to pass the string to the next item in the path.) Note that the user can pass an argument to the menu only when the session is opened for the first time.

```
MENU MAIN MKTMENU (&ARG
```

Accessing Text Files: Help, News, and Bulletin Boards

Text files can be quite helpful to your users. News files can keep them informed of events and issues in the company and department. Help text files can assist users with both the menus and the options listed on the menu.

Help and News

- 1. Create a help or news member using the PIE EDIT command, or upload an existing text member using the Repository Load Utility.**

See [See page 67, "Text Utility"](#). and [See page 77, "Repository Load Utility"](#). in the *Operation and Administration Guide* for procedures.

- 2. Then on the menu line, display it with BROWSE member-name.**

If you specify a Group ID, that group will be used. If you omit the Group ID, PIE/CICS conducts a standard hierarchical search for a matching member.

BROWSE MTKNEWS

Note: The Logon Director also has a news facility. Users can see company news before they log on.

Executing an Application in Another VTAM APPLID

Using PIE/CICS, you can log on to PIE/CICS on one VTAM application or CICS region and execute programs running in another. You can execute programs in different types of VTAM applications, such as TSO and IMS, and in unconnected CICS regions, such as stand-alone regions or multiple MRO TORs.

Using MultiCICS sessions to run programs in other VTAM applications helps create the image of a single, seamless system. Users don't have to remember what runs where. They just use sessions or menu lines to execute the programs they want.

You can make the process completely transparent to the user, which also enhances the single system image. Users can be automatically logged on to the target region, in which case the logon screen is not displayed. The application program is executed automatically. From the user's point of view, all the applications are executing from the same CICS region.

PIE/CICS handles the transfer of control to the target application and back to the originating application. It handles the sign on and can pass data, such as the PIE profile. It also handles errors, such as a target application being unavailable or a rejected logon. Instead of leaving the terminal at the VTAM logo, which could be confusing to a user who did not know he was passing through VTAM in the first place, PIE/CICS would return the user to his previous session—that is, the session he was in when he opened the VTAM session.

See ["Chapter 9 VTAM Applications" on page 109](#) of the *Customization Guide* for information on preparing execution strings.

Special Features of MultiCICS

Running a Native CICS Session

You can give users access to native CICS running in a MultiCICS session with the SYSTEM command. PIE/CICS restores the native transaction environment and presents the user with the CICS blank screen. Users can type in transactions as they would if you did not have PIE/CICS. We call this **system mode**.

PIE/CICS commands are still available in system mode (when prefixed with PEXE), and users can still switch between sessions with their escape string.

To give users access to system mode, specify SYSTEM in the **Execute** field.

☞ You cannot use AutoEnd with system mode. See "[Native CICS: System Mode](#)" on [page 27](#) of the *Operation and Administration Guide* for more information.

Using a Dynamic Menu as a Sessions menu

Normally, when a user selects an item from a PIE/CICS dynamic menu, the application executes in the same session as the menu. Often, however, MultiCICS customers want to run a menu in a single MultiCICS session and have its applications execute in other sessions. A dynamic menu can replace the Sessions menu.

This is helpful because dynamic menus are more flexible; they allow room for comments and customized header information. Also, they have menu security, which dynamically omits menu lines before a user sees the menu.

You can accomplish this using the LAUNCH command.

NetMizer Optimization

You may have to reset NetMizer optimization for particular applications. For example, a file transfer program may send data that is not in 3270 format. In this case, you must set NetMizer to NON3270. Refer to the *NetMizer Administrator Guide* for more information on NetMizer options.

Remember to code the NetMizer option before the command to be executed. If you code AE or another prefix, you must code NON3270 before the prefix. For example:

```
NON3270 AE CMD FLTR CICSTEST &ARG
```

Chapter 4 Case studies

This chapter consists of a series of case studies that show sample implementations of the steps outlined in [“Chapter 2 MultiCICS Customization” on page 5](#), and [“Chapter 3 Session Execution String” on page 23](#). These case studies present only an outline of what must be done. You will need to refer to the previous chapters for detailed instructions about these procedures. (Procedures will be cross referenced by page number.) For example, we do not describe how to get to the System Administrator menu in each one, because that is adequately described in [See page 5, “MultiCICS Customization”..”](#) Instead, the case studies focus on what to do after you have accessed a menu.

Case Study 1: System-Wide Implementation

This is the simplest of the case studies. All users need access to the same four applications. This case study shows how to use MultiCICS to give them four sessions, each with a pre-defined application. Users can switch between sessions using the escape string or the Sessions menu.

- 1. Access PIE/CICS as a system administrator and access the System Administrator menu (see [page 7](#)).**
- 2. Edit the User Directory (see [page 13](#)).**
- 3. Modify the model entry to suit your installation.**

Modify the system administrator entry (the first asterisk entry) to restrict PIE/CICS system administrator authority to only selected users. Customize the second asterisk entry as appropriate for your users. Set the Maximum Sessions field to 4. Set the Profile field to OURUSERS. You may leave the Groupid field blank.

The Terminal Directory is acceptable as shipped.

- 4. Create the OURUSERS profile using the Profile Utility (see [page 16](#)). Specify **SYSTEM** for the PIE profile group field, since you are using only one group.**

The default application is used when the user opens a session without specifying execution text. You will set up your system with all sessions pre-defined, so normally this will not occur. It can occur, however, if a user cancels a session and then reopens it. So it still makes sense to code an application here. You decide to specify a homegrown menu, so users can choose one of several applications:

Default Application: AE CMD ABCD

The initial PIE command is executed when the user first logs on to PIE/CICS. You want your users to see the Sessions menu, so they can select the session they need. You specify:

Initial PIE command: SESSIONS

The final PIE command executes when a user ends their last session. It is meant to keep the user from the blank CICS screen. It is never executed if the user goes to the Sessions menu. Since your users will be getting the Sessions menu when they log on,

this command will never be executed. You can blank it out, if you like, or code:

Final PIE command: EXIT LOGOFF

which, if it were executed, would log users off PIE/CICS and CICS and send them back to the VTAM logon screen.

Your profile has no special requirements on the Terminal or Keys screens. But, you do need to pre-define your sessions on the Sessions screen. You want the following four sessions on your Sessions menu:

```
PIE Sessions Menu ----- (C) TSC, Inc 1987
Command ==>
..#.. Name.... Title..... Esc Str: ==
_ 1 INVENTORY Inventory System      User: SD080BB
_ 2 ORDER      Order Entry System    Term: 0003
_ 3 CUSTOMER   Customer Information Database H Node: PIE10003
_ 4 EMAIL      E-Mail                          Sys : TSCDB
```

Keep in mind that the Sessions menu is not like a PIE dynamic menu. It does **not** omit lines from display if a user is unauthorized to execute the transaction defined for the session. Of course, your external security will continue to control the actual execution of the program.

To set this up, you code the Profile Sessions screen as follows:

```
PIE Profile Sessions Configuration ----- (C) TSC, Inc 1987
Command ==>
      Name.... Title..... Hide
1 Name ==> INVENTORY - Inventory System      -
Execute ==> AE CMD INVT

2 Name ==> ORDER      - Order Entry System    -
Execute ==> AE CMD ORDR

3 Name ==> CUSTOMER   - Customer Information Database - Y
Execute ==> AE CMD CINF

4 Name ==> EMAIL      - E-Mail                          -
Execute ==> AE CMD MAIL

5 Name ==>           -                               -
Execute ==>

6 Name ==>           -                               -
Execute ==>

PF: 1 Help 3 End 4 RETurn 7 UP 8 DOWN
```

Each transaction is prefaced with AE CMD. AE automatically ends the session when the user ends the application running in it. When you run a CICS transaction, you must run it with either CMD or TRAN (it defaults to TRAN if you don't code either). We use CMD because these transactions clear the screen before they execute, as most do. Using CMD, rather than TRAN, avoids an extra clear before the transaction executes.

Save your changes with PF5 and exit the Profile Utility.

Your MultiCICS system is set up. You can check it by signing on to PIE/CICS as a user. Enter PIE on a CICS blank screen and sign on with a user ID that does not have system administrator authority. You will see the Sessions menu. Open each session to ensure that it executes properly.



Case Study 2: VTAM Applications in Separate Sessions

Your general requirements are the same as for “ [Case Study 1: System-Wide Implementation](#)” on page 35. However, your users will be logging on to CICS1, and they need to use programs that run in other VTAM applications—an unconnected CICS region, TSO, and IMS. You want to pre-define those applications in the users’ profile.

1. **Access PIE/CICS as a system administrator and access the System Administrator menu (see [page 7](#)).**
2. **Edit the User Directory (see [page 13](#)).**
3. **Modify the model entry to suit your installation. Modify the system administrator entry (the first asterisk entry) to limit system administrator authority to you and, possibly, a few other technical users. Customize the second asterisk entry as appropriate for your users. Set the `Profile` field to `OURUSERS`.**

The Terminal Directory is acceptable as shipped.

4. **Create the `OURUSERS` profile using the Profile Utility (see [page 16](#)). Code values for:**

Default Application—`AE CMD ABCD`

When users open a session without specifying an execution text, PIE/CICS will execute transaction ABCD.

Initial PIE command—`PROC EC OPEN 1`

When users sign on, they will automatically go to session 1. Notice the PROC command. This is placed in the `Initial PIE command` field to process commands sent by another PIE/CICS region. You need this command because you have at least two unconnected CICS regions. You are certain that you will be passing control from this CICS region to another. You may also pass control from that CICS region to this. If no command is passed, whatever follows PROC will be executed. It makes sense to code PROC and a command for every initial PIE command when you have multiple PIE/CICS regions.

Also notice the EC prefix before the OPEN command. The OPEN command is a PIE Environment command, but the PROC command expects an Application command. The EC prefix allows an Environment command to be processed as an Application command.

Final PIE command—`EXIT`

When users end out of their last session, if they have not gone to the Sessions menu, or they end out of the Sessions menu when they have no open sessions, they will be presented with the Exit pop-up menu, which allows them to choose whether to disconnect, sign off, log off, or restart their PIE/CICS session.

Your profile has no special requirements on the Terminal or Keys screens. But you do need to pre-define your sessions on the Sessions screen. Your users need the following sessions on their Sessions menu:

```

PIE Sessions Menu ----- (C) TSC, Inc 1987
Command ==>
..#.. Name.... Title..... Esc Str: ==
_ 1 INVENTORY Inventory System User: SD080BB
_ 2 ORDER Order Entry System Term: 0003
_ 3 CUSTOMER Customer Information Database H Node: PIE10003
_ 4 EMAIL E-Mail Sys : TSCDB
    
```

To set this up, you code the Profile Sessions screen as follows:

```

PIE Profile Sessions Configuration ----- (C) TSC, Inc 1987
Command ==>
      Name.... Title..... Hide
1 Name ==> INVENTORY - Inventory System -
Execute ==> AE CMD INVT

2 Name ==> ORDER - Order Entry System -
Execute ==> ACCESS CICS2 DATA=&ZUSER/&ZPSWD ORDR

3 Name ==> CUSTOMER - Customer Information Database - Y
Execute ==> SCRIPT PLAY QMFCUST

4 Name ==> EMAIL - E-Mail -
Execute ==> REXX TSOEMAIL

5 Name ==> - -
Execute ==>

6 Name ==> - -
Execute ==>

PF: 1 Help 3 End 4 RETurn 7 UP 8 DOWN
    
```

Session 1 is coded with AE CMD, standard coding for transactions running in the same CICSplex.

Session 2 is coded with the ACCESS command, available only with NetGate. This command passes control to the target region, CICS2, while leaving the user signed on to the originating region, CICS1. The user can switch between all four sessions at any time.

Session 3 is coded with the SCRIPT command. Scripts are the best way to log users on to non-CICS regions automatically. There are two kinds of scripts: playback scripts and PIE/REXX scripts. SCRIPT PLAY executes playback scripts. You will create the script later in the implementation; it will log the user on to TSO and access QMF and the customer database.

Session 4 is coded with the REXX command. REXX executes PIE/REXX scripts. Later you will create a PIE/REXX script to log the user on to TSO and start E-mail.

Press PF5 to save your changes and PF3 twice exit the Profile Utility.

- To prepare your unconnected CICS region to receive the automatic signon and process the ORDR transaction, ensure that PIE/CICS is running in that region. Then edit the profile the user will sign on with. Its initial PIE command must begin with PROC. Type PROC before the command already in the field. If the command is a PIE Environment command (which is likely), code PROC EC before the command. Then press PF5 to save your changes and PF3 twice to exit.**



6. Create the playback script (see "Chapter 8 Scripts" on page 93 of the *Operation and Administration Guide*).

From the Sessions menu, enter `SCRIPT START ACCESS TSO`. This begins recording your script and starts the `ACCESS` command to TSO. TSO will ask you to enter your user ID. Enter the variable `&ZUSER`, so that the script will be appropriate for whoever executes it. Then TSO will present its logon panel and request your password. Enter the variable `&ZPSWD`. TSO will log you on and present you with the `READY` prompt.

Now it is time to execute the program you are after. In this case it is QMF. Enter `ISPF` and select QMF from your ISPF menus. Then enter the necessary commands to access your customer database.

Now it is time to end the script. Enter `==SCRIPT`. You will get the Script pop-up menu. Press `PF3` to stop the script. Press `ENTER` to return to the pop-up menu, then press `PF5` to save the script. Enter `SYSTEM.QMFCUST` as the script name. Press `ENTER` again, then `PA2` to exit from the Script pop-up menu.

To test your script, log off TSO. Then go to the Sessions menu and enter `SCRIPT PLAY QMFCUST`.

7. Create the REXX script (see the *PIE/REXX User Guide*). From the Sessions menu enter `EDIT SYSTEM.TSOEMAIL`. Enter the following:

```
/* PIE/REXX script to access E-mail in TSO */
OPTIONS 'SCRIPT=YES' /* Process program as script */
call Scr_Find '*', 'ENTER USERID- ', '...', 'I'
/* Find ENTER USERID field */
call Scr_Put_Field '*', ZUSER /* Put user ID in field */
'KEY ENTER' /* Enter input */
call Scr_Find '*', 'Password ====> ', '...', 'I'
/* Find Password field */
call Scr_Put_Password '*' /* Put password in field */
'KEY ENTER' /* Enter input */
ISPF EM /* Start ISPF and the EM pgm */
```

8. Press `PF3` to save the program and end the `EDIT` session.

9. Test your program. Log off TSO and enter `REXX TSOEMAIL` from the Sessions menu command line.

Your MultiCICS system is set up. You can check it by signing on to PIE/CICS as a user. Enter `PIE` on a CICS blank screen and sign on with a user ID that does not have system administrator authority.

Case Study 3: User Groups

In this example, users in several departments need entirely different applications. Each department needs its own profile, so that it can have its own applications pre-defined on the Sessions menu.

To do this you need to set up user groups. Normally, you would do this by defining generic user IDs in the User Directory and assigning each its own profile. But your user IDs are based on user names, not on department names. To set up different groups, you would have to create a separate User Directory entry for each user, which is too much work. So you want to tie your User Directory entries to your external security and use the groups defined in external security.

1. Access PIE/CICS as a system administrator and access the System Administrator menu (see [page 7](#)).

2. Edit the User Directory (see [page 13](#)).

Modify the model entry to suit your installation. Then modify the system administrator entry (the first asterisk entry) to limit system administrator authority to you and, possibly, a few other technical users. Then customize the second asterisk entry as appropriate for your users. Set the `Profile` field to a single profile name, such as `OURUSERS`.

In a few steps you will be implementing an exit to get the group name from RACF. So no Group name is needed. If you specified one, it would be overridden.

The Terminal Directory is acceptable as shipped.

3. Create multiple OURUSERS profiles using the Profile Utility (see [page 16](#)).

For each profile you will specify a different Group name in the `PIE profile group` field. Remember to use the Group names you have defined in RACF.

See "[Case Study 1: System-Wide Implementation](#)" on [page 35](#), for a sample profile.

4. The Post Initialization exit, PUXINIT, is the one to use to get Group names from RACF.

We provide a sample exit, `P04INIT`, in the macro library, that is set up to accomplish this. To put this in place, edit member `PCTTRACF` of the `SAMPLIB` dataset. Change `COPY PUXINIT` to `COPY P04INIT` and save your changes.

To reassemble the exit, edit member `PCTTPUX1` of the `CNTL` dataset. Make sure that the `SECURITY` parameter specifies `RACF`. Check other parameters to ensure that they are correct for your installation and run the job.

5. Verify that your MultiCICS system has been set up properly by signing on to PIE/CICS with different user IDs.

Enter `PIE` on a CICS blank screen and sign on with a user ID that does not have system administrator authority. Try user IDs from several groups to verify that you are getting different profiles for different user groups.

Chapter 5 Production and Maintenance

This chapter describes the steps you should take before placing MultiCICS into production. Ongoing maintenance is also discussed in this chapter.

Performance Considerations

The number of concurrent users and transactions are usually much higher in production than in a test environment. In production, PIE/CICS may have certain performance and tuning requirements that were not apparent during testing.

Before placing PIE/CICS into production you should:

- Review [“Chapter 11 Performance and Tuning” on page 153](#) of the *PIE/CICS Customization Guide*. Refer to the DSA section title [“Virtual Storage” on page 154](#), regarding putting CICS max class limitations on transactions PSGM and PCDP.
- Contact UNICOM Systems, Inc. Customer Service department regarding any HIPER PTFs for PIE/CICS

Making PIE/CICS the Good Morning Transaction

If you use the CICS VTAM good morning message program, you can bring up the Logon Director automatically on all PIE terminals. When users turn on their terminals, they can log on to PIE/CICS automatically.

If you do not make PIE/CICS the good morning transaction, users must enter the PIE transaction to access PIE/CICS.

For information on making PIE/CICS your good morning transaction, see the *PIE/CICS Customization Guide*, [“Signing On at the Logon Director Screen” on page 7](#).

Ongoing Administration

Little maintenance is required after you have set up a MultiCICS environment. There are several ongoing administrative issues that system administrators must consider as part of routine MultiCICS maintenance.

- When you apply maintenance to CICS or migrate to a new CICS release, you must re-assemble any customized system tables and exits.
- Changes in a site environment and personnel changes require routine, ongoing maintenance to PIE/CICS facilities that are used by MultiCICS. The following areas should be regularly updated to reflect those changes.

1. When new applications are added to your system or old applications removed, you may need to update user profiles.

If you have set up group administrators or allowed private object authority, that maintenance can be performed by users rather than by the system administrator. For more information on profiles, see [“Step 9: Set Up Profiles,” on page 16](#), and [“Chapter 3 Profiles” on page 29](#) of the *PIE/CICS Customization Guide*.

2. If your User Directory entries are specific (not simply an asterisk), entries must be added or deleted with the arrival and departure of users.

Because the User Directory is not accessible to group administrators, only system administrators can perform this maintenance. You can minimize User Directory maintenance by associating security updates to the User Directory. For more information, see [“ PUXINIT—Post Sign on Initialization” on page 149](#) of the *PIE/CICS Customization Guide*. For more information on the User Directory, see [“ User Directory” on page 52](#) of the *PIE/CICS Customization Guide*.

3. If your terminal entries are specific (not simply an asterisk), you must maintain Terminal Directory entries as well.

Like the User Directory, the Terminal Directory is accessible only to system administrators. We recommend that in most instances you set up an asterisk entry, to handle all terminals to reduce maintenance. For information on the Terminal Directory, see [“ Terminal Directory” on page 46](#) of the *PIE/CICS Customization Guide*.

4. You may have set up specialized control with user exits. As your needs change, these exits may have to be changed as well.

Exits are documented in the *PIE/CICS Customization Guide*.

Chapter 6 MultiCICS Basics

This chapter presents the basic operating concepts of MultiCICS and explains how to use commands and menus.

Since MultiCICS is so flexible and environments differ so widely, we will ask you to use the DEMO profile when you sign on to PIE/CICS and assume your environment is like that defined by the DEMO profile shipped with PIE/CICS. If you use a different profile, keystrokes and sessions may differ, but the overall concepts and processes will not.

Please read "[Chapter 1 MultiCICS Introduction](#)" on [page 1](#) before continuing. For concepts common to all PIE/CICS products, such as signing on and off PIE/CICS, see "[Chapter 2 Using PIE/CICS](#)" on [page 11](#) and "[Chapter 3 Commands](#)" on [page 21](#) of the *Operation and Administration Guide*.

Sessions Menu

The Sessions menu can be a center point for MultiCICS processing. From the Sessions menu, you can see which sessions you have defined and opened and switch to a new session. (We will discuss switching in the next section.)

You can also execute PIE/CICS commands and CICS transactions at the menu command line. Using the RECALL, or R, command, you can execute many similar commands quickly.

The Sessions menu is composed of several different screen areas.

- The header area consists of the top two lines of the screen.
The first line contains the screen title and copyright information. The second line includes a command line to enter PIE commands and CICS transactions.
- The bottom line of the screen shows the Sessions menu PF key definitions. See "[Sessions Menu Commands](#)" on page 45, for information on the menu commands.
- The environment information area is a column on the right side of the screen. Its fields identify the following.

Esc Str	Escape string (see " Escape String " on page 52)
User	User ID you signed on to PIE/CICS with
Term	CICS terminal you signed on with
Node	VTAM terminal node you signed on with
Sys	APPLID of the CICS system you are on
CPU	CPU you are running under
Ver	PIE/CICS version
Time	Time you accessed the Sessions menu (or pressed an AID key from the menu)
Date	Current date in mm/dd/yy and Day Month Date format.
Exit	Key to access the PIE/CICS Exit pop-up menu.

- The main portion of the screen is the session information area. Its fields are:
 - Line command area
 - # Session number ranging from 1 to 99
 - Name Session name followed by a one character status code. Possible codes are:
 - blank () Session is not active. It has been defined but not opened, or it has been closed and remains allocated.
 - * Current session. It is the last session you switched to.
 - + Session is open and allocated (permanent).
 - Session is open and free (temporary).
 - Title Session description
- The rightmost portion of the session information area, just before the environment information area, is the hidden status field. If a session is hidden, an H appears on this field.

- The Sessions menu can list up to 19 sessions on each screen. If you have more sessions available, an indicator appears beneath the information window at the right of the screen.

Sessions Menu Commands

There are several ways to enter commands from the Sessions menu.

- **Command line**

You can enter PIE Application commands and CICS transactions on the menu command line. If you enter these commands, PIE/CICS will open a new session and execute the command or transaction in that session. (We will say more about opening sessions later on.)

You can also enter PIE Environment commands at the command line. Environment commands affect your environment and your sessions. For example, there are commands to open, close, and switch to different sessions. (You do not have to prefix the Environment commands with EC.)

If you want to execute similar commands, one after another, on the command line, enter R (for recall) to redisplay the last command you entered.

- **Line commands**

There are several line commands. These commands execute PIE Environment commands that affect your sessions.

Enter line commands in the leftmost field in the session information area.

The following line commands can be entered from the Sessions menu:

A	ALLOCATE a permanent sessions
C	CLOSE the session.
D	DISPLAY information about the session.
F	FREE a temporary session
H	HIDE the session.
O	OPEN the session.
P	CANCEL the session.
S	SWITCH to the session.
U	UNHIDE the session.

- **PF keys**

Sessions menu PF key definitions are displayed at the bottom of the screen. If you are not authorized to use a command assigned to a PF key, that definition is not shown.

Help (PF1)	Access Help for the Sessions menu.
Open (PF2)	Open the session at the cursor position.
End (PF3):	End the Sessions menu.
Close (PF4):	Close the session at the cursor position.
Switch (PF5)	Switch to the session at the cursor position.
PIE (PF6)	Open the PIE pop-up menu.
View (PF10):	Open the View pop-up menu.
Bulletin (PF11)	Switch to the Bulletin board session.
Profile (PF12)	Access the Profile Utility.

Stacking Commands

You can specify multiple commands that are processed as a command string. You can type any combination of multiple line commands and commands on the command line (separated by a delimiter character, such as a semicolon).

When you press ENTER, commands are processed in the following order:

- All line commands are processed in session order, except for commands that cause a session switch.
- Commands are processed in the same sequence that they are entered on the Command line, except for commands that cause a session switch
- The first command in the command line that causes a switch will be processed. If there is no switching command on the command line, the first line command that causes a switch will be processed.

After a switching command has been processed, all other commands will be ignored. For example, if you specify the following stacked commands on your Sessions menu:

```
Command ==> TITLE 6 E-Mail; OPEN CEMT; FREE 6
```

The commands are executed in the following sequence:

- The C line command closes session 1.
- The line command U would unhide session 5. (The line command S would be skipped for now, because it causes a switch.)
- The command TITLE in the command line would change the title of session 6.
- The FREE command would remove session 6 from the menu. (The OPEN command would be skipped, because it causes a switch.) Now all non-switching commands have been executed.
- The OPEN command in the command line would open and switch to the next available session and execute CEMT in it.
- The S line command would be thrown away and session 3 would not be switched to, because the OPEN command caused a switch already and all subsequent commands are ignored.

Passing Session Arguments from the Sessions Menu

If a session accepts arguments, you can enter them from the Sessions menu when you switch sessions. To do so, enter the session by code or name followed immediately by the argument. For example, if session number 6 accepts an argument for the month, you enter the following on the command line:

```
6 JULY
```

MultiCICS opens session 6 and pass JULY as an argument. You can only pass an argument to a session when you open it. If the session is already open, the argument is ignored.

Types of Sessions

There are three types of sessions.

- The first type of session is called an `application session`. These are the standard sessions, numbered 1 to 99. You can run CICS applications, VTAM applications, PIE Application commands, and dynamic menus within these sessions.
- The Sessions menu runs in a special session named `SESSIONS` and the session number is 0.
- The third type is called a `special session`. These sessions are used by PIE/CICS to display your run-time profile, PIE pop-up menus, and View screens. They cannot be referenced by session number. These sessions end when you switch out of them.

Attributes of an Application Session

Application sessions have several potential attributes.

- **Open or Closed:** An open session is one that has been entered; the application in it is active. A closed session is defined but inactive; the application defined for the session is not running. To open a session that is already defined, enter its session number or name on the Sessions menu command line. See "[Ending Sessions](#)" on page 54, for information on closing a session.
- **Permanent (Allocated) or Temporary (Free):** A permanent or allocated session is displayed on the Sessions menu whether it is open or not. It stays on the Sessions menu when you close it. A temporary or free session is displayed on the menu only when it is open. When you exit it, it is removed from the menu. You can change this attribute with the `ALLOCATE` and `FREE` commands.
- **Hidden or Unhidden:** A hidden session is skipped when users execute the `SWITCH NEXT/PREV` command. An unhidden session is included in the `SWITCH NEXT/PREV` round. You can change this attribute with the `HIDE` and `UNHIDE` commands.

Defining Sessions

PIE profiles specify a set of sessions to be allocated when the user signs on to PIE/CICS. In addition, users can define new sessions with PIE Environment commands. The User Directory entry determines the maximum number of sessions a user can have defined at one time.

There are a number of ways to define new sessions.

The following commands define a permanent session, without switching to it.

- Use `NAME` and `ALIAS` to define a session and specify its name. The default application will be used both as execution text and title.
- Use `TITLE` and `NOTES` to define a session and specify its title. The default application specified in your profile will be used as execution text. The first word of the execution text will be assigned as the session name after you switch to the session.
- Use `CREATE`, `ALLOCATE`, and `ALLOCATH` to define a session and specify the execution text. If you omit the execution text, the default application specified in your profile will be executed. The execution text will be used for the title. The first word of the execution text will be assigned as the session name after you switch to the session.

- Use the START command to define a session and switch to it. You can specify an execution text. If you omit the execution text, the default application specified in your profile will be executed. The execution text will be used for both name and title.
- If you have private object authority, you can change your initial session configuration by changing your profile. To do so, see "[Changing User Profiles](#)" on page 18 of the *Operation and Administration Guide*. (You can change your run-time profile, even if you don't have private object authority. But because your changes will not be saved, it is usually easier just to define a session with one of the commands above.)

The following methods define a temporary session and switch to it.

- On the Sessions menu command line, enter an execution text (PIE Application command or CICS transaction). The execution text will be executed in the new session and will be used for both name and title.
- On the Sessions menu command line, enter the OPEN or ALLOCATT command. You can specify an execution text to execute in the session. If you omit the execution text, the default application specified in your profile will be executed. The execution text will be used as the title. ALLOCATT does not switch to the session. The first word of the execution text will be assigned as the session name after you switch to the session.
- From an input field in any session, enter your escape string followed by a transaction code or PIE Application command. The execution text will be executed in the new session and will be used for both name and title.

Example: Your escape string is == and you enter ==CEMT I TAS to execute the CEMT command in a new session.

☞ See "[Escape String](#)" on page 52 for more information on the escape string.

The session definition commands follow this general format:

<code>commandname [code] [execution text]</code>
--

If you specify a session code, MultiCICS will define the session you specify. If you omit it, MultiCICS will define the next available session.

If you specify an execution text, MultiCICS will execute that text in the session when you switch to it. If the text is not permitted by the command or if you omit it, MultiCICS will execute the default application in the session.

The execution text is not executed until you switch to a session. Sessions that were defined in your initial configuration or that were defined by issuing any of the first set of commands, above, are allocated but not open. Their execution texts will not run until you switch to those sessions.

When you use any of the second set of methods to define a session, you will switch to the session immediately. So the execution text will be executed immediately.

Multiple Commands

You can issue multiple commands to specify execution text, name, and title. For example, the commands—

```
ALLOCATE 10 AE CMD CEMT
NAME 10 CEMT
TITLE 10 Master Terminal Commands
```

—define session 10 to execute CEMT, name it “CEMT,” and give it an plain English title.

Exceeding the Number of Available Sessions

If you attempt to define a session when you have no more available sessions, you receive either the Switch pop-up menu or switch to the Sessions menu, depending on how your Customization Options are set. In either case, you will receive a message notifying you that you have no more available sessions.

Defining and Accessing Sessions: Exercises

Now it is time to try what you’ve learned in the preceding sections.

- 1. First make a note of your escape string. It is displayed in the environment information area on the right side of the Sessions menu. We will assume it is the default, ==. We will say more about the escape string later on.**

- 2. Open session 1 of your initial sessions configuration. Enter 1 on the Sessions menu command line.**

Session 1 will open and the application defined to it will start. When it has finished initializing, enter ==SES or ==0 on the command line to switch back to the Sessions menu. (The escape string must begin in the first position of an input field.)

- 3. Define a new permanent session. Enter ALLOCATE LIST USERS on the Sessions menu command line.**

The next available session, 5, will be created, with the execution text, LIST USERS, as the title. No name has yet been assigned.

- 4. Switch to the session. Enter 5 on the Sessions menu command line. You will see the PIE/CICS User List Utility. Switch back to the Sessions menu: enter ==0.**

- 5. Notice your new session’s name, USERS. You can give it a new name with the NAME command. Enter NAME 5 PIEUSERS.**

- 6. This session is a permanent session. That means that if you end the application running in the session, the session will remain defined on the Sessions menu.**

Let’s do this. Enter 5 on the Sessions menu command line. From session 5, press PF3 to end the User List Utility. You will switch back to the Sessions menu. Notice that the session is still defined on the menu.

- 7. The sessions that were predefined, sessions 1 through 4, are also permanent sessions. If you switch to them and end the applications running in them, the sessions will remain defined on the menu. You can re-enter them at any time.**

Try that with session 1. Enter 1 on the Sessions menu command line. Then press PF3 to end the menu and return to the Sessions menu. The session is still defined. Enter 1 again

on the command line to switch back to session 1. The same menu executes. Enter ==0 to return to the Sessions menu.

- 8. Let's define another permanent session. Enter `START LIST SCRIPT`. This will define the next available session, 6, switch to it, and execute `LIST SCRIPT`, the PIE/CICS Script List Utility. Switch back to the Sessions menu: enter ==0.**

Notice that session 6 was automatically assigned the name `SCRIPT`. Now switch back to session 6 (enter 6 on the Sessions menu command line), and press PF3 to end the Script List Utility. You will switch back to the Sessions menu.

Like session 5, session 6 ends but remains defined on the menu. Notice the difference, however: now session 6 has no name. That is because you left the name to be assigned, from the execute string, when you switched to the session. That is how the commands `TITLE`, `NOTES`, `CREATE`, `ALLOCATE`, `ALLOCATH`, and `START` work. In fact, if you executed something different in the session, the session would change its name.

- 9. Let's try that. Enter `START SYSTEM` in the command line. You will get to a blank CICS screen. At the blank screen, enter `PEXE LIST PROFILES`. This will start the Profile List Utility. Now switch to the Sessions menu (==0) and see that session 7, your new session, is named `PROFILES`.**

Return to session 7 (enter 7 at the Sessions menu command line). Press PF3 to end the utility and return to the blank screen. Now enter `PEXE LIST PANEL`. This will start the Panel List utility. Switch back to the Sessions menu (==0) and see that session 7 is now named `PANELS`.

Once again, look back at session 5 on the menu: you defined it with the `ALLOCATE` command, but it kept its name after the session ended. Why? Because you gave it a permanent name with the `NAME` command. Both `NAME` and `ALIAS` assign a permanent name—the name does not change and it does not disappear when the session is ended.

- 10. Now let's create a temporary session. Enter `OPEN LIST TEXT` on the Sessions menu command line, or enter just `LIST TEXT`.**

The next available session, 8, will be created and opened. The Text List Utility will execute in it. Switch back to the Sessions menu. Notice the session is defined on the menu.

- 11. Because session 8 is a temporary session, when you end the application running in it, the session will disappear from the Sessions menu.**

Let's try that. Enter 8 on the Sessions menu command line. From session 8, press PF3 to end the Text List Utility. You will switch back to the Sessions menu. Notice that the session is no longer listed on the menu.

- 12. Define more sessions, if you like. When you have finished, switch back to the Sessions menu.**

Switching Between Sessions

Now you have several sessions defined, and probably several more open.

Your open sessions are running in parallel. You can see only one of them at a time, and only the one you are looking at can perform processing. The other sessions are “suspended”.

To get to another session, you `switch` to it. You can think of sessions as channels on your TV and switching as changing channels. You can see only one channel at a time, but all channels are always available for viewing.

Because MultiCICS's primary function is to make it easy for you to get from one session to another, it provides several ways to switch. We'll discuss and practice the following switching methods:

- switching from the Sessions menu
- using the escape string
- using PF keys
- switching from an application menu

☞ You cannot switch out of a session in which an application is running—that is, performing processing. The exception is a NetGate session. You can switch out when an application is active and the application will continue to run while you are in other sessions.

What Happens During Session Switching?

When you switch out of a session, the current status and screen image of the application running in the session are saved. When you switch back to the session, its session status information is restored, its screen image is redisplayed, and the application running in it is given control. In other words, no data is lost when you switch out of a session. When you switch back, your session will be as if you never left.

When you switch out of a special session, like the View or Help session, the special session is closed.

☞ MultiCICS does not perform a 3270 read buffer when switching sessions, so switching overhead is low.

Sessions menu

To switch to a session from the Sessions menu, enter its session code or name on the menu command line. For example, to switch to session 1, enter 1 in the command line. To switch to a session named EDIT, enter EDIT in the command line.

You can also switch to a session using the S or O line commands. Tab down to the session you want and type S or O in the command field.

Escape String

The escape string is the most versatile switching tool in MultiCICS. You can use it to switch to any other session or to the Sessions menu. You can go directly to a session by identifying it or you can select a session from a pop-up menu. You can even use the escape string to execute a PIE/CICS command.

In the last section, you used the escape string to switch from a session to the Sessions menu.

When you enter the escape string in a session, you are passing control from your application to PIE/CICS. Depending on what you type after the escape string, PIE/CICS will switch you to another session, create a new session, manipulate sessions, and so on.

To enter the escape string, type it in any input field on a session screen. You must begin the string in the first position of the field. Then, with the cursor at the end of your escape string and its operands, press ENTER.

The escape string will not be saved as screen data, as long as it is entered correctly—that is, it begins in the first position of the field. PIE/CICS will read from the beginning of the escape string to the cursor and perform the processing you request. If the cursor is not in the field containing the escape string, the entire field is executed as a PIE Environment command.

Your escape string is defined in your profile. It can be changed with the PIE Environment command SET ESCSTR—the change will remain in effect until you log off.

Some applications will allow you to input only numerics on the screen. To use the escape string in this type of application, you must change it to + or - before you enter the application. If you access another PIE/CICS region using NetGate, you must change your escape string in one of those regions; if the escape string is the same in both, the higher level PIE/CICS will always receive control.

Identifying a Session with an Escape String

A quick way to switch from one session to another is to enter the escape string followed by the name or code of the destination session.

For example, to switch to the Sessions menu, type 0 or SES after the escape string: ==0 or ==SES

To switch to session 7, enter: ==7

To switch to a session named EDIT, enter: ==EDIT

You don't have to specify the session's full name. You can type only the first few characters. See "[Identifying Sessions](#)" on page 23 of the *Operation and Administration Guide*, for details.

Practice switching from session to session using the escape string and the session ID. You can practice with session codes 1 to 7 and session names APPLMENU, CICS, NEWS, NOTEPAD, PIEUSERS, SCRIPTS, and PANELS.

Finish in an application session.

If you specify a session name that does not exist, MultiCICS will open a new session and attempt to execute that name in the session as a CICS transaction. Type carefully!

Switching to an Unknown Session

If you want to switch to a session, but you don't know the ID, you can type ==SWITCH. The Switch pop-up menu appears with a list of currently defined sessions and their status. You can select a session by pressing the PF key associated with it in the menu. To get to the Sessions menu, press the CLEAR key.

1. **Try it now. Enter ==SWITCH.**
2. **Press PF2 to switch to session 2.**
3. **Enter ==SWITCH again.**
4. **The Switch menu displays only twelve sessions at a time—for twelve PF keys. To see more sessions, press ENTER.**
5. **Press CLEAR to switch to the Sessions menu.**

☞ You may have a PIE global key set to SWITCH. If you do, you can access the menu with that PF key.

Using PF Keys: The SWITCH Command

PF keys offer a convenient way to switch sessions. You can equate a PF key to a switch command, either in your profile or by using the SET command.

We recommend that you do not use PA keys for switching. Data typed on the session screen but not entered with an AID key will be lost if you switch with a PA key. This is a 3270 hardware limitation. When a PA1, PA2, or PA3 key is pressed, new screen data is not sent to CICS; so MultiCICS is unaware of this data and cannot save it.

The SWITCH command is versatile. You can set PF keys to:

- bring up the Switch menu
- switch to a few frequently used sessions by session name or code
- rotate through your sessions in numeric order
- switch back to the previous session

We have already discussed the first two options. Now we'll discuss the last two.

You can rotate through sessions in numeric order with the SWITCH NEXT and SWITCH PREV commands. These commands give you access to all your sessions with a single PF key.

You can switch back to the last session you were in with the SWITCH BACK command. This is useful when you want to toggle between two sessions.

1. **Set some PF keys now. Enter PROFILE from the Sessions menu to access your PIE/CICS profile. Enter 3 to access the Keys screen.**
2. **Select three PF keys. For one PF key, type SWITCH; for the next, type SWITCH NEXT; and for the third, type SWITCH BACK. Then press PF3 twice to exit the Profile Utility.**

You can specify an operand after NEXT or PREV to determine the kinds of sessions included in the rotation. See the SWITCH command in the *PIE/CICS Command Reference* for more information.

3. **Try out your new PF keys to see how they work. End at the Sessions menu.**

Ending Sessions

There are many instances in which the session will end automatically when you switch out of it or when you end the application running in it:

- Special sessions, like Help and View, end when you switch out of them.
- When you end a PIE/CICS menu or utility, the session will be ended automatically.
- When you run an ordinary CICS application with the AE (AutoEnd) command, the session will be ended automatically when you end the application.
- If the application uses the PIEEXEC API to issue an END command, the session will be ended automatically.

If you are running the application from a dynamic menu, the session will remain up until you end the dynamic menu. It will not end when you end the application.

Other times, you must end the session yourself:

- When you run an ordinary CICS application without the AE command, the session remains active until the application finishes executing. You return to the CICS blank screen. From there, you can end the session by doing either of the following:
 - Pressing the system end key defined in your profile.
 - Switching to the Sessions menu and issuing the CLOSE or CANCEL command—see “[Closing and Cancelling Sessions](#)” on page 55.
- If you enter an invalid CICS transaction at the blank screen, the session will end.
- From a session at the CICS blank screen (when the blank screen was not accessed from a dynamic menu), you can end the session using either of the methods named above: using the system end key or the CLOSE/CANCEL commands.

When you end a session, MultiCICS switches you to the last session you were in. Depending on how your Customization Options are set, that could be either:

- The last application session you were in or the Sessions menu, whichever you were in last
- The last application session you were in, always skipping the Sessions menu

If there is no eligible session remaining, PIE/CICS will execute the final PIE command.

Closing and Cancelling Sessions

In the previous section we used the words “cancelled” and “closed” and referred to the CANCEL and CLOSE commands. These terms and commands are not equivalent.

- A `closed` session remains defined and listed on the Sessions menu. You can re-start the session by switching to it.
- A `cancelled` session is removed from the Sessions menu. To re-start it, you must redefine it.

Permanent sessions are closed when you end them normally. Temporary sessions are cancelled when you end them. You can close a temporary session with the CLOSE command. You can cancel a permanent session with the CANCEL command.

You can cancel a session with the CANCEL command, whether the session is active or inactive. (Closing an inactive session does nothing; it remains inactive but defined on the Sessions menu.)

WARNING

When you issue CLOSE or CANCEL against a session with a transaction still running in it, MultiCICS will abend the transaction. All unsaved data is lost. You should always end any transaction before closing or cancelling the session that the transaction is running.

Some transactions are non-cancellable. A transaction will be non-cancellable if it is conversational and has set a CICS abend exit—that is, has issued a EXEC CICS HANDLE ABEND command. A transaction can also be designated non-cancellable in the PIE/CICS CANTABLE.

In addition, some users can be denied cancel authority. These users will not be allowed to cancel any sessions. Cancel authority is specified in the User Directory entry, in the `Allow Cancel` field.

If you issue the CLOSE or CANCEL command but you have been denied cancel authority or you issue it for a non-cancellable transaction, MultiCICS will issue an error message, telling you that you must end the application normally.

There are several ways to issue the CANCEL and CLOSE commands.

- You can issue CLOSE and CANCEL from a session using the escape string—for example, `==CANCEL id`, where `id` is the name or code of the session to cancel. To end the session you are in, you can use an `*` as the session ID—for example, `==CLOSE *`.
- From the Sessions menu command line, enter `CLOSE id` or `CANCEL id`.
- From the Sessions menu, enter the `C` (close) or `P` (cancel) line commands next to the sessions you want to end.
- From the Sessions menu, place the cursor on the session you want to end and press PF4, CLOSE.
- If you don't identify a session—that is, you issue just `==CLOSE` from a session or CANCEL from the Sessions menu or you press PF4 key on the menu without placing the cursor on a session—a pop-up menu will appear. Press the PF key corresponding to the session you want to end. The current session will be displayed in high intensity.
- From a dynamic menu or API, execute `EC CLOSE id` or `EC CANCEL id`.

Exercises

You already ended a temporary session in “[Defining and Accessing Sessions: Exercises](#)” on [page 49](#). Now it is time to close and cancel some of your permanent sessions.

- 1. Close session 1. Tab down to the line command field next to session 1. Enter C in the field.**

Notice that the + status code disappears, but the session remains on the menu.

- 2. Cancel session 2. Tab down to the line command field next to session 2. Enter P in the field.**

Notice that it disappears from the menu.

- 3. Close session 5 from the Sessions menu command line. Enter CLOSE 5 at the command line.**

Again, the + status code disappears but the session remains on the menu.

- 4. Cancel session 6 from the command line. Enter CAN 6 at the command line.**

Session 6 disappears from the menu.

- 5. Go to session 3. Enter 3 on the command line. Enter ==CLOSE on the command line.**

You will get the CLOSE pop-up menu. Press PF3 to close session 3.

- 6. Go to session 7. Press PF3. The application ends, but the session does not. That is because you started session 7 with the SYSTEM command and thereby entered “system mode.”**

Now that you have ended the Panel List Utility, you are in system mode again. To end out of system mode, you must press a system end key. With the DEMO profile, the system end keys are PF3 and PF15.

Press PF3 now to end out of system mode in session 7. The session is a permanent session and is now closed. (The + status code no longer appears on the menu.)

Appendix A Customer Support

This appendix describes how to get help from Customer Service when you experience a problem with a UNICOM Systems, Inc. product. This appendix includes separate sections that describe several diagnostic suggestions to rule out user errors and the information you should have ready before reporting the problem.

Contacting Customer Service

UNICOM Systems, Inc. Customer Service can be reached by the following methods:

Voice	818-838-0606
Fax	818-838-0776
Email	support@unicomsi.com

A Support and Services web page provides Customer Service information about all of UNICOM Systems, Inc. products. Use the following URL to browse the Support and Services web page:

<http://www.unicomsi.com/support/index.html>

The Support and Services web page provides an online form to report a problem with a UNICOM Systems, Inc. product. Use the following URL to complete and submit a Technical Support Request form:

<http://www.unicomsi.com/support/>

Normal business hours are from 6:00 a.m. to 4:30 p.m. Pacific Standard Time, Monday through Friday. Emergency customer service is available 24 hours a day, 7 days a week.

An answering service receives customer service calls beyond normal business hours. You may leave a message if it is not an urgent problem. A customer service representative will return your call at the start of the next business day.

Requests for urgent support outside of normal business hours are answered immediately. A customer service representative will be summoned to return your call. Leave a phone number where you can be reached. If you have not received a return call from a Customer Service representative within an hour of reporting the problem, please call back. Our customer service representative may be experiencing difficulties returning your call.

International customers should contact their local distributor to report any problems with a UNICOM Systems, Inc. product.

Troubleshooting Suggestions

This section describes several troubleshooting suggestions to diagnose common errors that can cause PIE/CICS problems. Before calling Customer Service, follow these suggestions to rule out the possibility these errors are causing your PIE/CICS problem.

- Run the Installation Verification Program with the **P#IV** transaction. Browse the PIECIVP temporary storage queue to see if it contains error messages that suggest problems with the allocation of CICS programs, transactions, maps, or files.
- Verify any recent changes to your site's operating system, CICS, or other products are fully compatible with PIE/CICS.
- Verify all load modules are at the same release level if a new release of PIE/CICS was installed over a previous release.
- Check that all modules were reassembled after upgrading PIE/CICS or applying maintenance to CICS.
- Verify all PIE/CICS system tables were reassembled after applying maintenance to CICS or upgrading to another release.
- Verify all users have current PIE/CICS passwords and entered them correctly.
- Examine your CICS logs, MVS console, and PIE/CICS logs for error messages from not only PIE/CICS, but any other product that runs concurrently with PIE/CICS.

Describing the Problem

Gather the following information about your system environment before reporting a problem to UNICOM Systems, Inc. Customer Service:

Operating system release and PUT Level _____

VTAM system release and PUT Level _____

PIE/CICS release _____

Date of PIE/CICS distribution tape _____

Gather the following information about your CICS system before reporting a problem to UNICOM Systems, Inc. Customer Service:

CICS release and PUT Level _____

CICS configuration MRO/ISC etc. _____

Real or virtual terminal _____

Before calling UNICOM Systems, Inc. Customer Service, get answers to the following questions.

What PIE/CICS products were active when the problem occurred?

Availability Plus

Dynamic Menus

MultiCICS

NetGate

NetMizer

NonStop CICS

Is the problem occurring in the TOR or AOR? _____

Is the problem occurring in a production or test region? _____

What is the severity of the problem? _____

What are the major symptoms of the problem? _____

Is the problem re-creatable under specific conditions? _____

Has the problem occurred more than once? _____

Were changes made to CICS or PIE/CICS immediately prior to the occurrence of the problem? _____

What other software products were running when the problem occurred? _____

Is a diagnostic message produced when the problem occurs? If so, what is the ID and text of the messages? _____

Does an abend occur? If so, what are the abend and return codes? _____

Is a dump produced when the problem occurs? If so, what kind of dump is it? _____

Please try to be as accurate and complete as possible in answering these questions. Your problem can be resolved more quickly if a customer service representative has all of the pertinent information needed to find a solution.



Appendix B. System Tables

System tables allow you to customize various elements of the PIE/CICS environment. The tables are fully documented in the *PIE/CICS Customization Guide*. This appendix describes the tables that apply to MultiCICS.

☞ You must re-assemble the system tables when you apply maintenance to CICS or migrate to a new CICS release.

- CANTABLE: Cancel Table

Use the CANTABLE table to prevent particular applications from being cancelled. PIE/CICS forces users to end all CANTABLE applications normally. If the user is to be logged off the system (for example, by PASS or if their User Directory entry specifies DISC=NO and they enter DISC) and an application in this table is running, they will be disconnected, not logged off. If they attempt to cancel one of these applications, they will get an error message stating that they must end the application normally.

- TERMASK: Terminal Mask Table

The TERMASK table is optional. PIE/CICS creates pseudo CICS terminal IDs to make the terminal IDs for each session unique. Use the TERMASK table to determine how PIE/CICS creates the pseudo terminal ID.

- TSQFLDS: Temporary Storage Fields Table

Use the TSQFLDS table to define the CICS fields that will be used by applications as part of a temporary storage name, for example, the terminal ID and operator ID. Since the same temporary storage queue name can be written at the same terminal by more than one session, each session must use a physically unique name.

- USERFLDS: User Fields Table

Use the USERFLDS table to define the CICS control block areas that must be unique for each PIE session, not shared across different sessions. For example, the TCTTE user area cannot be shared.

Appendix C Exits

PIE/CICS exits are fully documented in the *PIE/CICS Customization Guide*. This appendix gives a description of only those exits that affect MultiCICS. Refer to the *PIE/CICS Customization Guide* for information about all other exits.

Session Shutdown Exit

The Session Shutdown exit, PSCPCAN, is invoked whenever a PIE session is cancelled or closed.

A transaction running in a session that was cancelled cannot do clean up work. Cancelling a session causes the same abnormal end of the transaction as a CICS lost session. You can execute the logic in your DFHZNEP in the Session Shutdown exit, to perform clean up processing for cancelled sessions.

PUXVIEW, View Command Access Exit, and PUXVSCN, View Screen Access Exit

Use the View Command Access exit, PUXVIEW, to control who can access PIE/CICS View. Use the View Screen Access exit, PUXVSCN, to control who can view a particular screen.



Appendix D Application Program Interfaces

PIE/CICS APIs are fully documented in the *PIE/CICS Customization Guide*. This appendix describes only those APIs that pertain to MultiCICS.

PIEEXEC API

You can execute PIE Environment commands using the PIEEXEC API. To do so, prefix the command with EC or ENV.

TCTTE User Area API

Extract or update the TCTTE User Area for switched out PIE sessions. Use this API for clean up processing at logoff time in DFHZNEP or in the PIE/CICS Shutdown exit. See member PCSSMUAA of the MACLIB dataset for more information.

Sessions Menu API

You can change the title (description) of a session on your Sessions menu using the Sessions menu API.

- 1. To use this interface, link to program PC@UPTIT with a commarea containing the new title, which can be up to 44 characters.**
- 2. You can examine the session's title before you change it. To do so, use the PIE variables API, PC@UPVAR, and check the &ZTITLE variable.**

For more information on the PIE variables API, see >>Variables API in the *Command Reference*.

- 3. The API must be running in the session whose title you want to change.**



Appendix E Limiting Transactions to One Session

You can restrict certain transactions or groups of transactions to execute in a single session. If a user attempts to execute a transaction or member of a transaction group in another session, PIE/CICS displays a message to the user that the transaction cannot execute in this session.

☞ PIE/CICS must be initialized in AOR regions if you are using MRO.

To implement this feature, do the following.

- 1. Go to the Customization Options screen.**
- 2. Specify YES for the Limit Transactions to One Session field.**
- 3. Edit member PCSSTPCP of the SAMPLIB dataset.**

PCSSTPCP is a tranid directory with ordered entries similar to the User and Terminal Directories. Like these other directories, you can use wildcards to designate multiple entries. The default table looks like the example shown in the following figure:

```
*-----*
*                GROUP   TRANID  LIMIT
*                NAME    MASK    FLAG
*-----*
          DC   CL4'      ',CL4'PEXX',C'N' DO NOT LIMIT TRANID 'PEXX'
          DC   CL4'      ',CL4'PEDT',C'N' DO NOT LIMIT TRANID 'PEDT'
          DC   CL4'      ',CL4'*    ',C'N'  LIMIT ALL OTHER TRANIDS.
*-----*
```

The tranid of a started transaction is checked against this table. A sequential search is made from the beginning of the table. After a matching tranid is found, the “Limit Flag” field is checked to see if it is:

- Y Limit the tranid to one session
- N Do not limit the tranid to one session
- H Limit to one session for transactions that pseudo-converse using the “hidden tranid” method (the tranid is the first thing read from the screen and it causes that tranid to be initiated)

The default table, shown above, limits no tranids to one session. If the Limit Flag on the last (*) entry were changed from N to Y, then the table would limit all tranids, except for PEXX and PEDT, to one session (the * entry matches all tranids).

To group transactions, code the same group name in the Group Name field for all tranids you want grouped together.

For example, the following table limits the billing tranids BL01 BL02, and BL03—as a group—to one session. Only one transaction belonging to group BILL can be running at a time. If BL01 or BL02 or BL03 is active in PIE session 2, then BL01, BL02 or BL03 cannot be started in any other PIE session.

Appendix E Limiting Transactions to One Session

The table limits all other tranids, except PEXX and PEDT, to one session each.

```
*-----*
*                GROUP   TRANID  LIMIT
*                NAME    MASK    FLAG
*-----*
DC   CL4'BILL',CL4'BL01',C'Y'  Limit billing group
DC   CL4'BILL',CL4'BL02',C'Y'  Limit billing group
DC   CL4'BILL',CL4'BL03',C'Y'  Limit billing group
DC   CL4'      ',CL4'PEXX',C'N' DO NOT LIMIT TRANID 'PEXX'
DC   CL4'      ',CL4'PEDT',C'N' DO NOT LIMIT TRANID 'PEDT'
DC   CL4'      ',CL4'*      ',C'Y'  LIMIT ALL OTHER TRANIDS.
*-----*
```

- 4. Assemble table PCSSTPCP using JCL in member PCSSTPCP of the CNTL dataset. Changes to the table become effective the next time you cycle CICS.**

Index

A

- accessing other VTAM applications 11
- accessing PIE as an administrator 7
- Application automatic end detection 26
- attributes of application sessions 47
- audience vii
- Avg No. of Sessions per User field 8

B

- blank CICS screen 3
- book conventions xi

C

- cancel table 61
- cancelling a session 55
- CANTABLE table 55, 61
- CICS
 - blank screen 33
 - running a native CICS session with MultiCICS 33
- closing a session 55
- commands
 - LAUNCH 33
 - Sessions menu line commands 46
 - stacking 46
 - SYSTEM 33
- compatibility with other software products 12
- contacting Customer Service 58
- Customization Options facility 8– 9

D

- default application 17
- Dynamic Menu
 - use with MultiCICS 33

E

- escape string 52– 53

F

- final PIE command 18

G

- good morning transaction 41

I

- initial PIE command 17

L

- LAUNCH command 33
- limiting transactions to one session 67– 68
- Logon Director 2, 22

M

- maintenance 42

N

- NetMizer optimization 34

P

- P#IV transaction 58
- PCSSTPCP table 67
- Performance 41
- performance considerations 41
- Phone and fax numbers, UNICOM Systems, Inc. Software 57
- PIE Exec
 - API 65
- PIECVP TS queue 58
- PIEEXEC API 65
- planning 6
- profiles
 - defining PF keys 19
 - defining toggle key 19
 - maintenance 42
 - setting default application 17
 - setting final PIE command 18
 - setting initial PIE command 17
 - using Profile utility 16
- PSCPCAN exit 63
- PUXVIEW 63

PUXVIEW exit [63](#)
PUXVSCN exit [63](#)

R

recommended reading [ix](#)
Repository [10](#)

S

scripts [3](#)
session
 cancelling [55](#)
 closed [47](#)
 closing [55](#)
 defining [47– 50](#)
 ending [54](#)
 hidden [44, 47](#)
 open [47](#)
 permanent [47](#)
 switching [51](#)
 switching to an unknow session [53](#)
Session cancel exit [63](#)
Sessions menu
 commands [45– 46](#)
 description [2](#)
 format [44](#)
 line commands [45](#)
 passing arguments [46](#)
 switching sessions [51](#)
 types of sessions [47](#)
Sessions Menu API [65](#)
sessions shutdown exit [63](#)
switching
 By Pf Key [53](#)
syntax conventions [xi](#)
SYSTEM command [33](#)
System mode
 leaving [26](#)
system mode [33](#)
system tables [61](#)

T

TCTTE User Area API [65](#)

Temp Storage Name Substitution [8](#)
TERMASK table [61](#)
Terminal Directory [15, 42](#)
toggle key [19](#)

transactions
 P#IV [58](#)
transferring screen data between sessions
 [3](#)
TSQFLDS table [61](#)
types of sessions [47](#)

U

User Directory [13– 14, 42](#)
USERFLDS table [61](#)