# CICS Comet

Installation and Administrator Guide

Release 5.4.0

First Edition printed September 2016



This manual applies to CICS Comet release 5.4.0



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

# Table of Contents


**ABOUT THIS MANUAL ..... IV**

AUDIENCE ..... IV
CUSTOMER SERVICE ..... V

**CHAPTER 1 OVERVIEW OF CICS COMET ..... 1**

INTRODUCTION TO THE FEATURES OF CICS COMET ..... 1
*API and Non-API Macro Programs ..... 1*
*Use CEDF to Debug Macro-Level Programs ..... 1*
*Virtual Storage Constraint Relief ..... 1*
*Run Macro-Level Programs in MRO ..... 2*
*Subsystem Storage Protection (SSP) ..... 2*
*Transaction Isolation ..... 2*
*CICS Comet Avoids Source Code Converter Pitfalls ..... 2*

**HOW CICS COMET WORKS ..... 3**

RESOURCE ISSUES ..... 4
*CICS Release Compatibility ..... 6*

**CHAPTER 2 INSTALLATION ..... 7**

INSTALLATION CHECKLIST ..... 7
*Step 1: Unload the First File from the Product Tape ..... 8*
*Step 2: Copy CICS Comet Distribution Libraries to Disk ..... 9*
*Step 3: Install CICS Comet into Your CICS system ..... 10*
*Creating PPT and PCT Entries ..... 10*
New Users ..... 10
Existing Users ..... **Error! Bookmark not defined.**
*PLT entry required for CICS Comet ..... 11*
PLACE CICS COMET AFTER — ..... 13
PLACE CICS COMET BEFORE — ..... 13
STEP 4: START-UP OVERRIDES ..... 14
STEP 5: UPDATING CICS START-UP JCL ..... 15
STEP 6: INITIALIZING DYNAMIC ATTACH ..... 16
STEP 7: DYNAMIC ATTACH FOR MACRO-LEVEL TRANSLATION ..... 17
Selecting Programs to Attach ..... 17
Matching ..... 17
Displaying Included and Excluded Programs ..... 17
Making Dynamic Changes to Tables ..... 17
Different Tables for Different Regions ..... 18
CCMACINC INCLUDE TABLE ..... 19
MIXED ..... 19
RMODE ..... 20
FIX RMI ..... 20
FIX CSA ..... 20
FIX CBL ..... 20
*CCMACEXC Exclude Table ..... 21*
Example Entries within the CCMACEXC Exclude Table ..... 22

*Step 8: Additional PPT Entries* ..... *23*
*Step 10: Verifying Installation (Optional)* ..... *25*

**CHAPTER 3 CUSTOMIZATION** ..... **27**

*Customizing Other Software for Use with CICS Comet* ..... *28*
*InterTest, XPEDITER, or ADSP Debuggers* ..... *28*
*PL/I* ..... *28*
CUSTOMIZING CCSIPARM ..... 30
*CCSIPARM Parameters* ..... *32*
*Password Parameters* ..... *37*
MOVING BMS MAPS ABOVE THE 16 MB LINE ..... 38
*Sample Relink for BMS Mapsets -SAMPLIB(BMSMAPS)* ..... *38*
RUN MACRO LEVEL PROGRAMS ABOVE THE 16 MB LINE ..... 39
COMMAND LEVEL PROGRAMS—CSA ADDRESS ..... 40
Listing Included and Excluded Programs ..... 40
Making Dynamic Changes to Tables ..... 40
Different Tables for Different Regions ..... 40
*CCCMDINC Include Table* ..... *41*
Example ..... 41
*CCCMDEXC Exclude Table* ..... *42*

**CHAPTER 4 DEBUGGING CICS COMET** ..... **43**

USING CEDF WITH CICS COMET ..... 45
*Example of a Macro Level Request and Its Command Level Equivalent* ..... *46*
Macro Level Request ..... 46
Command Level Equivalent Under CEDF ..... 46
*Summary* ..... *47*
*Finding the Abending PSW and Registers* ..... *49*
CICS COMET ASRA ABEND FACT SHEET ..... 52
*Information About the Abending Program* ..... *52*
DECODING STORAGE OPERANDS ..... 53
PROGRAM DEBUGGING - OVERVIEW ..... 54
*CICS Comet Domains* ..... *54*
Exit Points Within CICS Comet Domains ..... 54
*Environment Built to Support Macro Level* ..... *55*
How CICS Comet Gets Control ..... 55
CSA/CWA ..... 55
TCA/TWA ..... 55
PCT ..... 56
PPT ..... 56
TCTTE ..... 56
FCT ..... 56
THE CICS COMET STUB AND THE PROGRAM OFFSET ..... 56

**CHAPTER 5 SPECIAL PROCEDURES** ..... **57**

RELINKING MACRO-LEVEL PROGRAMS ..... 58
EXAMPLES OF RELINK JCL ..... 59
COBOL Macro-Level: SAMPLIB(CBLMACRO) ..... 59
COBOL Mixed-Mode: SAMPLIB(CBLMIXED) ..... 59
PL/I Macro-Level: SAMPLIB(PLIMACRO) ..... 60
PL/I Mixed-Mode: SAMPLIB(PLIMIXED) ..... 60
Assembler Macro-Level: SAMPLIB(ASMMACRO) ..... 61
Assembler Mixed-Mode: SAMPLIB(ASMMIXED) ..... 61
INSTALLING INTERFACES ..... 62
*IDMS* ..... *62*
*TOTAL/DB* ..... *64*

SYSTEM 2000 (S2K) DATABASE—R11.6 ..... 65
SUPPORT FOR COBOL II AND DATABASES ..... 66
CODING ENTRIES FOR FUNCTION SHIPPING ..... 67
*Coding Remote DCT Entries ..... 67*
*Coding Remote FCT entries ..... 67*

SUPPORTING DIRECT CICS CONTROL BLOCK SEARCH ..... 69
*Coding Non-XA FCT entries ..... 69*
*Coding Non-XA DCT entries ..... 70*
*Coding Non-XA JCT entries ..... 71*
*Coding Non-XA PCT entries ..... 72*
*Coding Non-XA PPT entries ..... 73*
*Coding Non-XA TCT entries ..... 74*
*Coding Non-XA TST entries ..... 75*
ISAM COMPATIBILITY ..... 76
CICS COMET EXITS ..... 77
*How to Use the Exits ..... 80*
**Appendix A. About CICS Comet** ..... 81
RESTRICTIONS AND STATEMENT OF SUPPORT ..... 81
Messages and Codes ..... 83
Common Abend Codes ..... 90
Comet SAMPLIB Members ..... 92
SAMPLE COBOL PROGRAM TO ACQUIRE CSA ADDRESS ..... 94
**Appendix B. Commands and Debugging** ..... 96
CICS COMET COMMANDS ..... 96
HALY Service Transactions ..... 99
*CICS Comet's EIS Trace Table ..... 107*
**Appendix C. Performance Monitoring** ..... 115
**Appendix D. CICS Comet Program Stub** ..... 117
Index ..... 118

# About This Manual

This manual describes how to install and customize CICS Comet Release 5.4.0. The manual also includes typical administrative procedures to maintain CICS Comet after it has been fully implemented.

## Audience

This manual is intended for system administrators and operators responsible for CICS operations at their site. Readers are expected to understand MVS, OS/390, z/OS, and CICS concepts. Also, many of the procedures described in this manual require site-specific JCL changes. Readers should be proficient editing JCL and familiar with their site's system standards.

# Customer Service

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

Voice: 818-838-0606
Fax: 818-838-0776
Website: http://www.unicomsi.com/support
Email: support@unicomsi.com

Normal business hours are from 7:00 a.m. to 4:00 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 representative may be experiencing difficulties returning your call.

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

# Chapter 1 Overview of CICS Comet

This chapter consists of two major sections. The first section introduces the major features of CICS Comet. The second section describes how CICS Comet converts macro-level programs.

## Introduction to the Features of CICS Comet

With CICS Release 3.1, IBM removed support for PL/I and COBOL macro-level programs. With CICS Release 3.2, IBM dropped support for all CICS programs that use macro-level calls. With CICS release 3.3, IBM removed access to the CSA control block. In migrating programs to TS versions of CICS, you must contend with the difficulties and expense of finding, changing, converting, and testing source code of all existing CICS programs that use macro-level calls. Surveys of data centers have produced estimates of $250,000 to $1,000,000 to convert CICS source code. Third party software packages present even greater difficulties because the source code is controlled by the vendor.

You can defer or eliminate the trouble and expense of migrating macro-level programs with CICS Comet. CICS Comet intercepts all macro-level calls (including CICS system level calls, such as CTYPE=LOCATE) and dynamically translates them to the proper command level calls. Your existing macro-level programs can run in CICS/TS regions without modifications. With CICS Comet, you are able to take advantage of improvements in CICS that are part of the TS versions of the product.

#### API and Non-API Macro Programs

CICS Comet handles all standard API macro-level programs, and in many cases also handles non-standard API macro-level programs. Many older macro-level programs use non-standard API techniques that may require adjustments to CICS Comet or the use of an exit. If you discover a program that does not operate correctly with CICS Comet, we will be happy to work with you to get it to operate correctly. Sometimes this requires an interactive process between you and our technical development group (sending information to us and making adjustments to CICS Comet parameters or assembling a CICS Comet exit). Our technical staff has yet to find a macro-level application program that can’t run under CICS Comet.

#### Use CEDF to Debug Macro-Level Programs

Using CICS Comet, you can take advantage of CEDF debugging for your macro-level programs. Cometized macro-level programs can also use data tables and remote FCT, DCT, and temporary storage queues.

#### Virtual Storage Constraint Relief

CICS Comet provides virtual storage constraint relief by moving programs and BMS mapsets to the XA address space where they can be executed and accessed without inefficient cross-memory or memory to memory moves.

- CICS Comet allows macro-level programs to run above the 16 Mb line without source changes. This includes VS/COBOL, PL/I, and some assembler programs.
- CICS Comet automatically moves all BMS maps above the 16 Mb line.
- CICS Comet allows you to specify DATALOC(ANY) on both the PCT and PPT of macro-level programs, running 31-bit Cometized application code.

☞ To run macro-level programs or to allocate their storage requests above the line, you must specify `RMODE=Y` in the CCMACINC table.

**Run Macro-Level Programs in MRO**

IBM's MRO (MultiRegion Operation) does not support function shipping or data tables if your programs are macro-level. You cannot use File Owning Regions and Data Owning Regions.

CICS Comet allows any macro-level program to run using CICS MRO without changes to the program, even if you run CICS 2.1. Placing transactions and applications into AORs offers improved performance and reliability with the use of MRO and data tables.

**Subsystem Storage Protection (SSP)**

VS/COBOL programs using COBOL verbs, such as INSPECT and UNSTRING, will abend with OC4 ASRA using SSP in CICS 3.3. However, using CICS Comet solves this problem; programs do not abend when using SSP.

**Transaction Isolation**

CICS Comet fully supports applications capable of running with transaction isolation (TRANISO).

**CICS Comet Avoids Source Code Converter Pitfalls**

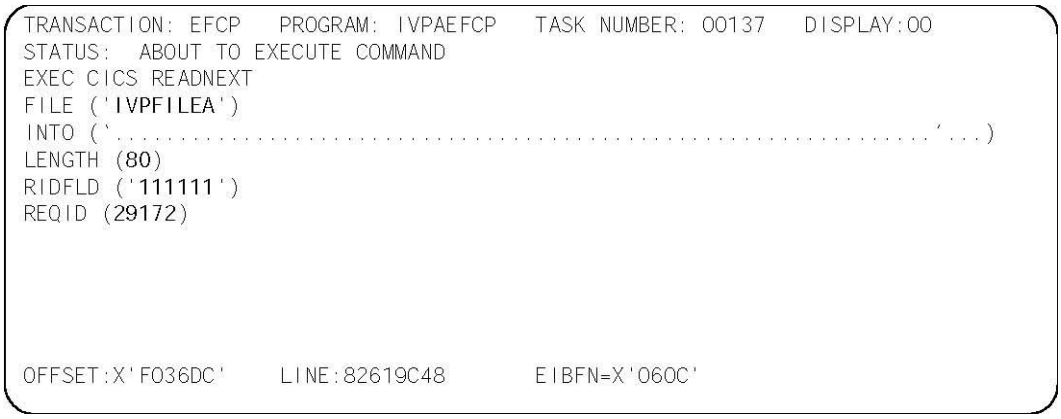
The problem with source code converters is that not all the information you need to complete the command level request is present when you encounter the macro-level request in a source program.

For example, the source code you are trying to convert does not know the file name. The macro-level request does not fill out the file name (FILE), key field address (RIDFLD), I/O area length (LENGTH), I/O area address (INTO), or request ID (REQID). For example, the length comes from the FCT, which is not available to source code translators, or in the assembled FCT and the file name comes from the FCT pointer in the file work area. The program simply has the command:

```
DFHFC TYPE=GETNEXT,ERROR=BROWSE_ERROR
```

The source code converter cannot translate this into an executable command.

In contrast, CICS Comet dynamically places the file name, length, key field, I/O area, request ID, and return address into the command level call.




# How CICS Comet Works

CICS Comet adds a small stub (similar to the CICS command level stub, DFHEAI, DFHECI, or DFHEPI) to every program it translates. You could relink the program to include the stub, but relinking poses serious operational issues. Relinked programs cannot run in versions of CICS below 3.1 without CICS Comet. Until you move all programs to CICS/TS, you must maintain two libraries. When changes are made to the source code, you must make changes to both libraries. CICS Comet eliminates relinking issues by dynamically attaching the stub to programs. No relinking is necessary.


Unlike CICS Comet, some software programs use the CICS XPCFTCH exit to dynamically attach themselves to user programs. XPCFTCH is called each time you use a program, whether the program is dynamically attached or not. On a typical system, the number of program uses per day is extremely high—five, ten, fifteen million per day. On average, you have 4.5 times more transactions run than program fetches from a load library. In addition, processing is substantial each time the exit is called. Dynamically attaching with XPCFTCH imposes a severe performance overhead on your system.

In contrast, CICS Comet uses its own method to attach the CICS Comet stub. Comet Dynamic Attach is called each time the program is loaded, not every time the program is run. The number of program loads per day is quite low as compared to the number of times the program is run. In fact, many programs are loaded only once per day. Because CICS Comet performs Dynamic Attach infrequently, it has little if any impact on system performance.

When you specify a program for Dynamic Attach, you can indicate the program is mixed-mode (has both macro and command level code). You can also tell CICS Comet to move and execute the program above the 16M line in CICS/TS.

BMS mapsets are automatically moved above the 16M line. Assembler programs or programs with assembler subroutines that are not 31 bit compatible (use the high order byte of addresses to store data or operate as flags) cannot be moved above the 16M line.

# Resource Issues

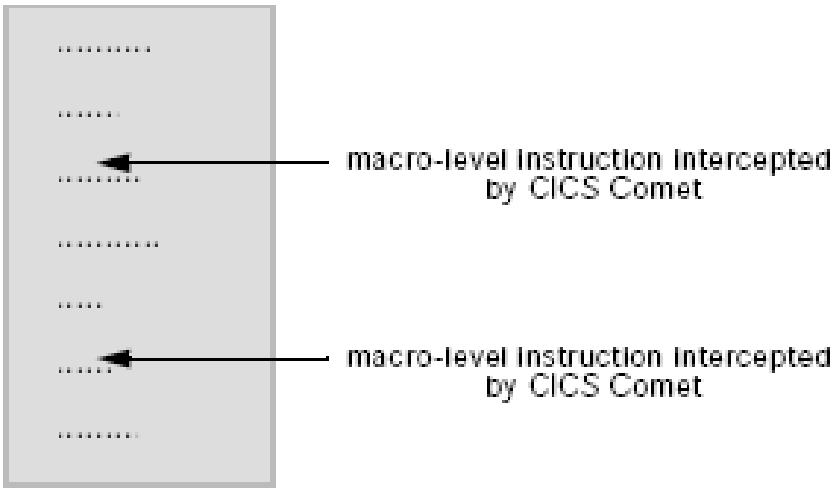
CICS 2.1 internally translates all command level code to macro-level code before execution. However, CICS/TS accepts only command level code, and executes that code directly. (This is one of the reasons behind the IBM statement that you will see a performance improvement for your command level programs under CICS/TS.)

If you run CICS Comet under CICS 2.1, your programs are translated twice. First, CICS Comet translates the macro code to command level; then CICS translates the command level code back to macro-level before execution.

The amount of translation overhead is dependent upon the amount of macro-level code in the program. Resource usage is minimal if your program's execution path is 30,000 instructions, but there are only two macro-level calls.

But, even for programs that require more macro code translation, you should not experience a significant impact on performance. For the most part, transaction response time is more dependent upon file I/Os than CPU processing. The overall improvements provided by CICS/TS should more than compensate for any overhead attributable to macro-level translation.

30,000 Instruction Application program



**CICS Release Compatibility**

CICS Comet determines which release of CICS you are running at start-up time. It then prepares a CICS 2.1 virtual environment for your applications to operate. Any CICS control blocks are copied to a separate area in CICS 2.1 format that are accessed by your applications.

CICS Comet's architecture minimizes the impact of any changes to CICS from release to release. Most of CICS Comet's work is translating macro-level code to command level code, which does not change much between releases. You can use the same version of CICS Comet on any supported release of CICS/TS with minimal changes.

CICS Comet fully supports application code from as early as CICS 1.5. This 5.4.0 release of CICS Comet can operate in the following CICS releases:

- Transaction Server 4.1 - (CICS 6.6.0)
- Transaction Server 4.2 - (CICS 6.7.0)
- Transaction Server 5.1 - (CICS 6.8.0)
- Transaction Server 5.2 - (CICS 6.9.0)
- Transaction Server 5.3 - (CICS 7.0.0)

# Chapter 2 Installation

This chapter describes how to install CICS Comet. The chapter consists of a series of numbered steps. Each step includes a procedure to complete a specific part of the installation process. After you complete all of the steps in this chapter, proceed to “Customization”, beginning on page 27 to complete the procedures to adapt CICS Comet to your system environment.

CICS Comet supports all CICS TS releases from Version 6.2.0 through 6.4.0

## Installation Checklist

This checklist summarizes the installation steps described in this chapter. Update this checklist as you complete each step to monitor your progress through the installation process

___1. Unload the first file from the CICS Comet distribution tape. This file creates a CNTL data set containing the INSTALL member.
___2. Run the INSTALL procedure in the CNTL data set to unload the remaining CICS distribution libraries from tape to disk.
___3. Add CICS Comet to your CICS system's PPT, PCT, and PLT tables
___4. Add start-up overrides.
___5. Update CICS start-up JCL.
___6. Set up the Dynamic Attach initialization procedure.
___7. Customize the distributed Include and Exclude tables for Dynamic Attach to include and exclude macro-level and mixed-mode programs from CICS Comet processing.
___8. Ensure that any macro-level assembler programs that will be dynamically attached by CICS Comet are defined dynamically through the use of CEDA and your DFHCSD dataset.
___9. Restart your CICS region and start CICS Comet.
___10. Verify CICS Comet was installed correctly.

# Step 1: Unload the First File from the Product Tape

The example IEBCOPY job shown below is used to unload the first file from the CICS Comet product tape.

1. Customize the job card and the SYSUT1 DD UNIT=CART parameter.

   Change the DSN and VOL=SER parameter to match the version of CICS Comet on the product tape.

2. Change the SYSUT2 DD DSN parameter to match the version of CICS Comet you received and add a VOLSER to the statement.

```
//JOB@NAME      JOB   1,UNLOAD,CLASS=A,MSGCLASS=X,REGION=2048K
//CNTL          EXEC  PGM=IEBCOPY
//SYSPRINT      DD    SYSOUT=*
//SYSIN         DD    DUMMY
//SYSUT1        DD    DSN=CCV540.CNTL,                  <== VERSION NUMBER
//                    LABEL=(01,SL,EXPDT=98000),
//                    VOL=SER=CCV540,                   <== VOLSER NUMBER
//                    DISP=OLD,UNIT=CART, <== TAPE UNIT ADDRESS
//                    DCB=(RECFM=FB,BLKSIZE=4080,LRECL=80)
//SYSUT2        DD    DSN=CICS.COMET.CCV540.CNTL,       <== VERSION NUMBER
//                    DISP=(NEW,CATLG,DELETE),
//                    DCB=(RECFM=FB,BLKSIZE=4080,LRECL=80),
//                    SPACE=(CYL,(1,1,47)),
//                    UNIT=SYSDA,VOL=SER=??????         <== VOLSER NUMBER
```

   You may also want to change the UNIT=SYSDA.

3. Run the job and check for condition code 0.

## CICS Comet Tape Format

File 1: CCV540.CNTL

File 2: CCV540.MACLIB

File 3: CCV540.SAMPLIB

File 4: CCV540.LOAD

File 5: CCV540.OPTIONAL

File 6: CCV540.ASMIVP

File 7: CCV540.COBOLIVP

File 8: CCV540.PLIIVP

File 9: CCV540.LOADPTFS

**Step 2: Copy CICS Comet Distribution Libraries to Disk**

The INSTALL procedure is located in the CICS.CCV540.CNTL dataset created when the first file was unloaded from the CICS Comet product tape.

1. Customize the job card and the volume parameters before submitting the INSTALL job.

   Add values to the TAPE, INDEX, DASD, and VOLUME parameters that match your site's standards.

```
//JOB@NAME JOB 1,INSTALL,CLASS=A,MSGCLASS=X,REGION=2048K
//*-------------------------------------------------------------------
//INSTALL PROC VERSION=CCV540,
//         TAPE=CART,              <=== UNIT ADDRESS FOR TAPE DRIVE
//         DASD=SYSDA,             <=== UNIT ADDRESS FOR DISTRIBUTION LIBRARIES
//         VOLUME=??????,          <=== DASD VOLUME TO ALLOCATE LIBRARIES ON
//         INDEX='CICS.COMET'      <=== HIGH LEVEL INDEX FOR DATASET NAMES
//*-------------------------------------------------------------------
```

   ☞ If you make INDEX= more than one node, the dataset name must be enclosed within quotes. For example, `INDEX='CICS.COMET'`

2. Run the INSTALL procedure and check for condition code 0 on all eight steps.

   For the last step (LOADPTFS ),you can get a valid condition code of 8 if the file is empty.

The remaining IEBCOPY datasets on the CICS Comet product tape are cataloged as:

- *hlq*.CCV540.MACLIB (File 2)
- *hlq*.CCV540.SAMPLIB (File 3)
- *hlq*.CCV540.LOAD (File 4)
- *hlq*.CCV540.OPTIONAL (File 5-8)
- *hlq*.CCV540.LOADPTFS (File 9)

☞ The IVP datasets on the product tape are merged into the OPTIONAL dataset when the INSTALL procedure is executed. The OPTIONAL dataset can be scratched when you have finished testing CICS Comet.

Do not use members from the LOADPTFS dataset unless instructed to do so by a representative from UNICOM Systems Customer Service.

### Creating PPT, PCT and FCT Entries

This step includes separate procedures for new and existing users. Use the procedure shown below if you are a new user. Use the procedure on the page 11 if you are migrating from an older to newer version of CICS Comet.

☞ The following UPGRADE jobs upgrade CICS Comet into CICS. They do not upgrade CICS. You must run one of these jobs to install CICS Comet.

## New Users or Existing Users

The following sample JCL is located in the COMET.CCV540.SAMPLIB(ADDTOCSD). This job is used to manage your CICS DFHCSD dataset with PPT, PCT and FCT entries needed by CICS Comet.

1. Customize the job card and the group name(COMETCSD).

```
//COMETCSD EXEC PGM=DFHCSDUP
//STEPLIB  DD DISP=SHR,DSN=SYS4.CICSTS??.CICS.SDFHLOAD    <=??=RELEASE
//SYSPRINT DD SYSOUT=*
//DFHCSD   DD DISP=SHR,DSN=YOUR.DFHCSD.VSAM.DATASET.NAME  <=DFHCSD
//SYSIN    DD *
*-------------------------------------------------------------------*
*      CICS/COMET TRANSACTIONS
*-------------------------------------------------------------------*

DEFINE TRANS(HALL)    GR(COMETCSD) PROG(CC$$STRT)   TASKDATAKEY(CICS)
       TASKDATALOC(BELOW)

DEFINE TRANS(HALE)    GR(COMETCSD) PROG(CC$$SNAP)   TASKDATAKEY(CICS)
       TASKDATALOC(ANY)

DEFINE TRANS(HALY)    GR(COMETCSD) PROG(CC$$SNAP)   TASKDATAKEY(CICS)
       TASKDATALOC(ANY)

DEFINE TRANS(HAKC)    GR(COMETCSD) PROG(CC$$ECAT)   TASKDATAKEY(CICS)
       TASKDATALOC(BELOW)          TWA(12288)

*-------------------------------------------------------------------*
*      CICS/COMET PROGRAMS
*-------------------------------------------------------------------*

DEFINE PROG(CC$$LAST) GR(COMETCSD) LANG(A) DATALOC(ANY)   EXECKEY(CICS)
       RES(NO)        USAGE(TRANSIENT)

DEFINE PROG(CC$$REUS) GR(COMETCSD) LANG(A) DATALOC(ANY)   EXECKEY(CICS)
       RES(YES)

DEFINE PROG(CC$$SOCC) GR(COMETCSD) LANG(A) DATALOC(ANY)   EXECKEY(CICS)
       CO(THREADSAFE) API(OPENAPI) RES(NO) USAGE(TRANSIENT)

DEFINE PROG(CC$$STRT) GR(COMETCSD) LANG(A) DATALOC(BELOW) EXECKEY(CICS)
       RES(NO)

DEFINE PROG(CC$$SNAP) GR(COMETCSD) LANG(A) DATALOC(ANY)   EXECKEY(CICS)
       CO(THREADSAFE) API(OPENAPI) RES(NO)
```

```
DEFINE PROG(CC$$XOCC) GR(COMETCSD) LANG(A) DATALOC(ANY)   EXECKEY(CICS)
       CO(THREADSAFE) API(OPENAPI) RES(NO)

*--------------------------------------------------------------------*
*      CICS/COMET IVP FILE
*--------------------------------------------------------------------*

DEFINE FILE(IVPFILEA) GR(COMETCSD) LSR(NONE) DSNS(ALLREQS) STAT(ENABLE)
       DISP(SHARE) STRING(3) DATABUF(4) INDEXBUF(3) RECORDF(F)
       ADD(YES) BROWSE(YES) DELETE(YES) UPDATE(YES) READ(YES)
       DSNAME(CICS.COMET.CCV540.IVPFILEA.CL)        OPEN(STARTUP)
/*
//
```

2. Change the STEPLIB and DFHCSD DD statements to match your site's naming conventions.
3. Submit the COMETCSD job.

☞ **This is the Minimum amount of resources that need to be defined, and the remaining PPT resources will be done the first time using EXEC CICS CREATE commands. This makes an Upgrade to a new release of Comet more self defining or automatic concerning PPT Resources.**

**PLT entry required for CICS Comet**

CICS Comet can be started automatically by placing CC$$STRT in the start-up PLT, or from any CICS 3270 type terminal.

The following is a sample start-up PLT (PLTPI) to start CICS Comet. This sample can be found in member DFHPLTCC of the SAMPLIB dataset.

☞ For CICS Release 3.2 and above, you must have an entry for DFHDELIM in the PLTPI as a delimiter between PHASE 1 and PHASE 2 PLTPI processing. (See below).

```
          DFHPLT TYPE=INITIAL,SUFFIX=CC
*-------------------------------------------------------------------*
* DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM  STG 2 DELIMITER
*-------------------------------------------------------------------*
DFHPLT TYPE=ENTRY,PROGRAM=CC$$STRT  BEGIN CICS/COMET STARTUP
DFHPLT TYPE=ENTRY,PROGRAM=CC$$SOCC  START CICS/COMET O/C EXIT

PLACE YOUR PLT ENTRIES HERE

*-------------------------------------------------------------------*
   DFHPLT TYPE=ENTRY,PROGRAM=OMOCICS  OMEGAMON MUST FOLLOW COMET
*-------------------------------------------------------------------*
   DFHPLT TYPE=ENTRY,PROGRAM=EISET  OMEGAMON MUST FOLLOW COMET
*-------------------------------------------------------------------*
   DFHPLT TYPE=ENTRY,PROGRAM=ACF2STRT  START ACF2
*-------------------------------------------------------------------*
*    THE FOLLOWING CICS COMET PLTPI ENTRY SHOULD BE DEAD LAST
*-------------------------------------------------------------------*
   DFHPLT TYPE=ENTRY,PROGRAM=CC$$LAST  FINISH CICS/COMET STARTUP
*-------------------------------------------------------------------*
DFHPLT TYPE=FINAL
END DFHPLTBA
```

✓ This statement is required in CICS/TS to force initialization of the VS/COBOL interface prior to starting CICS Comet.

✖ The CC$$SOCC PLT program enables the CICS Comet CC$$XOCC global user exit at the XFCSREQ exit point. The program ending in SOCC goes in the PLT, whereas the program ending in XOCC is the actual G.L.U.E. The CC$$SOCC entry is only necessary if all of the following conditions are true:

- You are running CICS/TS
  AND
- You specify FCT=SUFFIX in CCSIPARM
  AND
- You create a pseudo FCT that specifies FILSTAT=(CLOSED) or is left to the default.

For more information, see both the FCT and OPEN parameters in “Customizing CCSIPARM” on page 30, and “Supporting Direct CICS Control Block Search” on page 70.

## PLACE CICS COMET AFTER —

Place CICS Comet after DFHDELIM and all third party software (except Omegamon/CICS).

## PLACE CICS COMET BEFORE —

For all versions of CICS, place CICS Comet before all macro-level applications. Start CICS Comet before Omegamon/CICS—place it before OMOCICS and EISET. Also you must start CICS Comet before ADS (which is normally started from a terminal).

# Step 4: Start-Up Overrides

Specify `DISMACP=NO i`n the SIT or SIP overrides to make testing easier for mixed-mode programs running in CICS/TS regions.

# Step 5: Updating CICS Start-up JCL

This step explains how to concatenate the CICS Comet load library to the CICS DFHRPL. You must concatenate the Comet library after adding PPT, PCT, and PLT entries and before restarting CICS.

1. Add the CICS Comet load library to the DFHRPL concatenation.

```
//DFHRPL   DD DISP=SHR,DSN=CICSTS??.CICS.SDFHLOAD     CICS loadlib
//         DD DISP=SHR,DSN=hlq.CCV540.LOAD            Comet load library
```

   Make sure the CICS Comet load library is concatenated in the DFHRPL before the InterTest load library. The load library must be concatenated before the InterTest load library because Comet supplies sample InterTest user exits called IN25UEXI and IN25LETX.

2. Add the following DD statements:

```
//COMETPDS  DD DISP=SHR,DSN=hlq.CCV540.SAMPLIB     COMET samplib dataset
//COMETLOG  DD SYSOUT=*                            Dynamic Attach log
//COMETMAC DD SYSOUT=*                             Macro statements log
//COMETCSA  DD SYSOUT=*                            Exec CICS ADDRESS CSA log
```

# Step 6: Initializing Dynamic Attach

This step initializes the Dynamic Attach facility. The CC$$V540 job starts the Dynamic Attach facility. The following example shows CC$$V540, which is located in the SAMPLIB dataset.

```
//JOB@NAME JOB 1,CC$$V540,CLASS=A,MSGCLASS=X,REGION=2048K
//*-------------------------------------------------------------------
//*      SAMPLE JCL TO START THE CICS COMET DYNAMIC ATTACH FOR CICS
//*      VERSION 4.1 AND ABOVE.  THE LOADLIB THAT CC$$V540 IS EXEC-
//*      UTED FROM MUST BE APF AUTHORIZED.  IT IS OUR SUGGESTION THAT
//*      YOU MOVE CC$$V540 TO AN APF AUTHORIZED LIBRARY AND THEN
//*      CHANGE THE STEPLIB IN THE FOLLOWING JCL TO POINT TO THAT
//*      LIBRARY INSTEAD OF THE COMET DISTRIBUTION LIBRARY.
//*-------------------------------------------------------------------
//*      WHEN RUNNING CICS COMET IN A PRODUCTION ENVIRNOMENT IT WOULD
//*      ADVISABLE TO RUN THIS JCL AFTER EVERY MVS IPL.  THIS MAY
//*      MEAN CONVERTING THIS TO A PROC THAT CAN BE STARTED FROM
//*      THE MVS "COMMAND" MEMBER IN SYS1.PARMLIB.
//*-------------------------------------------------------------------
//CC$$V540    EXEC    PGM=CC$$V540
//STEPLIB     DD      DISP=SHR,DSN=&INDEX.CCV540.LOAD
//SYSPRINT    DD      SYSOUT=*
//SYSUDUMP    DD      SYSOUT=*
//SYSIN       DD      DISP=SHR,DSN=SYS2.PARMLIB(CC$$V540)
//
```

- You must execute the CC$$V540 job before Using Cics/Comet.

You can invoke CC$$V540 by the following methods:

- Include CC$$V540 in the z/OS IPL procedure (i.e., start a PROC with CC$$V540 in it). This is the recommended method.
- Execute CC$$V540 from the console as a procedure.
- Include CC$$V540 in your DFHSIP JCL (*only recommended for initial testing*).

Complete the following procedure to execute CC$$V540.

1. Authorize the CICS Comet LOADLIB or copy CC$$V540 to an APF authorized library.
2. Build a procedure containing CC$$V540 if you are going to include CC$$V540 in your IPL procedure, or execute it from the console.
3. Do one of the following:
   - Add the new CC$$V540 procedure to your IPL procedure.
   - Issue `S 'procname'` from the console to execute the CC$$V540 job.
   - Place the distributed CC$$V540 JCL in front of the execution statement for DFHSIP.

☞ ( *In the CICS JCL, is not recommend for anything but initial testing!* )

# Step 7: Dynamic Attach for Macro-Level Translation

The CICS Comet stub must be attached to every program it translates. CICS Comet dynamically attaches the stub when the program is loaded. You select the programs to dynamically attach with entries in the CCMACINC Include and CCMACEXC Exclude tables.

## Selecting Programs to Attach

You can locate the majority of your macro-level and mixed-mode programs using the IBM MSCAN utility. (Note that MSCAN will not find every program, and may occasionally incorrectly identify a program.) The CICS Comet stub should be added to that program and to all other programs in that transaction.

In some cases you cannot use Dynamic Attach to attach the CICS Comet stub; you must relink the program. See "Relinking Macro-Level Programs" on page 58, for more information.

## Matching

A program must be specified in the CCMACINC Include table to be dynamically attached. You can insert wildcards in the Include table to generically select a range of programs by a common name element. An asterisk (*) indicates any character in that position and forward. A question mark (?) indicates a single character in that position only.

Because of wildcard matches, there may be some programs placed into the Include table that should not be processed by CICS Comet. To exclude these programs, specify them in the Exclude table. Wildcards can be used to select entries for the Exclude table also. For example, if the Include table has an entry of PAY*, but you want CICS Comet to ignore programs beginning with PAYR*, then put an entry for PAYR* in the Exclude table.

An Exclude table entry overrides an Include table entry, unless the Include table entry specifies all eight characters of the name without wildcards. For example, if you have PAY* in your Include table and PA* in your Exclude table, all program names beginning with PA are excluded. However, PAYR1234 would be included because all 8 characters of the program name have been specified. Therefore, it is best not to put entries in the Exclude table that are too generic.

CICS Comet checks the Include table before the Exclude table. In both tables, CICS Comet uses the first match it finds. As a result, you should enter the most specific entries first in the tables, followed by your generic wildcard entries.

## Displaying Included and Excluded Programs

When CICS Comet is active, you can display the list of included macro-level programs by entering HALLEY LIST INCLUDED MACRO PROGRAMS or HALLEY L I M from any CICS terminal. To display the macro-level programs that have been excluded, enter HALLEY LIST EXCLUDED MACRO PROGRAMS or HALLEY L E M.

Both commands list the programs in the order they are entered in the table. If the list exceeds one page, "MORE…" is displayed at the bottom right corner of the screen. Press ENTER to display the next page of programs. Press CLEAR to exit.

## Making Dynamic Changes to Tables

You can change Include/Exclude table program entries at any time from ISPF. To refresh the tables in memory, enter HALLEY REFRESH MACRO from any CICS terminal. CICS Comet performs a NEWCOPY and reloads the programs in memory. The CICS Comet stub is either

attached or removed from the program in memory, based upon the changes made to the Include and Exclude tables. For example, if you add the OSC* entry to the Include table, CICS Comet issues a CEMT SET PROGRAM(OSC*) NEW for these programs.

## Different Tables for Different Regions

You can use different Include and Exclude tables for different CICS regions. You can name the tables whatever you like. However, all tables must be stored in the Comet SAMPLIB dataset specified in your CICS start-up JCL.

Complete the following procedure to use different tables for different regions:

1. Set up different CCSIPARM members for different regions.

   See "Customizing CCSIPARM" on page 30, for details.

2. Specify Include and Exclude table names with the PROGRAM_INCLUDES and PROGRAM_EXCLUDES parameters in the CCSIPARM member controlling that region.

# CCMACINC Include Table

The following example shows an excerpt from the CCMACINC member located in the SAMPLIB dataset. CCMACINC is used by CICS Comet to include macro-level programs for translation to command-level.

```
/*------|-|-|-|-|-|-|-|-|---------------------------------------
/*      | |  |F|F|F|-|-|-|
/*      | |  |I|I|I|-|-|-|
/*      |M|R|X|X|X|-|-|-|
/*      |I|M|  |  |  |-|-|-|
/*      |X|O|R|C|C|-|-|-|
/* PROG |E|D|M|S|B|-|-|-|
/* NAME |D|E|I|A|L|-|-|-|  PROGRAM DESCRIPTION...
/*------|-|-|-|-|-|-|-|-|---------------------------------------
IVPA$*  | | | | | | | |  INCLUDE ALL ASM IVP MACRO-LEVEL PGMS
IVPC$ALL| | | | | | | |  INCLUDE IVP CBL MACRO-LEVEL PROGRAMS
IVPC$MNU| | | | | | | |  INCLUDE IVP CBL MACRO-LEVEL PROGRAMS
IVPC$BRW| | | | | | | |  INCLUDE IVP CBL MACRO-LEVEL PROGRAMS
```

Enter each program name in the PROG NAME field, beginning at column 1.

### MIXED

A program is mixed-mode if it is a command-level program that:

- Is called from a macro-level program
- Contains macro-level calls.
- Contains EXEC CICS ADDRESS CSA and expects the CWA to follow the CSA, +512 bytes.
- Contains EXEC CICS ADDRESS TWA and expects the TCA to precede the TWA, -256 bytes.
- Contains EXEC CICS ADDRESS CSA or obtains BLL cell TCACBAR from CSACDTA.
- Doesn't GET CSACBAR from EXEC CICS ADDRESS CSA. For instance, it uses a VS/COBOL BLL cell 3 back door to get the CSA address.
- Calls an Assembler subroutine that issues macro-level requests.
- Calls macro-level database subroutines.

Enter a value in the MIXED field for each entry in the Include table:

**Y** YES for a mixed-mode program. Y is the default.
N NO to indicate that a program is not mixed-mode. For true command-level programs (i.e., not mixed-mode), N in the MIXED field improves performance. For true macro-level programs, it is ignored.
2 COBOL II Language Environment enabled programs.
L PL/I Language Environment enabled programs.


WARNING

Do not use * (asterisk) as a generic entry to include all programs without also customizing an Exclude table with entries for all third party system software, like CICS monitors, debuggers, dump products, dynamic FCT, label allocation products, and other system software.

IDMS command-level programs must be considered mixed-mode if they use the IDMS 10.2 interface. This is because the IDMS 10.2 database interface uses macro-level requests, even though the application program is completely command-level. System 2000 command-level programs must be considered mixed-mode if they use the System 2000 macro-level database interface. TOTAL/DB command-level programs must also be considered mixed-mode if they are using the TOTAL/DB interface.

## RMODE

This field applies only to CICS/TS. Place an * in the RMODE column to designate that COBOL II programs should remain where they were link edited, but should be dynamically attached by CICS Comet. For Assembler, PL/I, and VS/COBOL programs, an * entry is treated as an N. (See below.)

For specific VS/COBOL and PL/I application programs, place Y for Yes in the RMODE field to specify the program should be run above the 16 Mb line. Do not put a Y in the RMODE field until the program has been tested below the line (AMODE(24) RMODE(24)) and verified that if works correctly with CICS Comet.

Place an N for No to designate that the program should run below the 16 Mb line. N is the default—blank is assumed to be No. This is necessary for Assembler programs and subroutines that incorrectly use the high order byte of the address (necessary to run the program above the 16 Mb line) as flags or indicators. Unpredictable results could occur if CICS Comet relocated these programs above the line. That is why Assembler programs must be manually relinked to be RMODE=31 unless you set ASM-RMODE=YES in CCSIPARM. (An * indicates N for Assembler programs.)

☞ On rare occasions, a COBOL II program that is linked to run in 31-bit mode will need to be Cometized. In CICS/TS, you can use * in the RMODE column to handle the program.

## FIX RMI

Specify Y to indicate this program uses DB2 RMI calls. If the program does not use DB2 RMI calls, specify N to improve performance. Y is the default

## FIX CSA

If Y, CICS Comet will scan for direct CSA LOCATE macros. If this program does not contain these macros, specify N to improve performance. Y is the default.

## FIX CBL

If N, CICS Comet will not scan for and fix the DFHCBLI stub in mixed-mode VS/COBOL programs. Y is the default.

☞ FIX RMI, FIX CSA, and FIX CBL require some additional CPU cycles at program fetch time. To determine whether they are needed, set them to Y. Then, check your console log for CICS Comet messages. If you do not receive messages CCLL191I or CCLL193I for a program, you can set FIX RMI to N. If you do not receive messages CCLL192I or CCLL194I for a program, set FIX CSA to N. If you do not receive messages CCLL196I or CCLL198I for a program, set FIX CBL to N.

## Example

```
/*      | | |F|F|F|-|-|-|
/*      | | |I|I|I|-|-|-|
/*      |M|R|X|X|X|-|-|-|
/*      |I|M| | | |-|-|-|
/*      |X|O|R|C|C|-|-|-|
/* PROG |E|D|M|S|B|-|-|-|
/* NAME |D|E|I|A|L|-|-|-|  PROGRAM DESCRIPTION...
/*------|-|-|-|-|-|-|-|-|---------------------------------------
S2K*    |Y| | | | | | |  Mixed-mode program
ACPX*   | |Y| | | | | |  Run this program above the 16M line
PAY001P | |Y|N| | | | |  Don't scan for RMI calls.
CBL2PROG|Y|*| |N| | | |  Don't scan for CSA LOCATE macros.
```

**CCMACEXC Exclude Table**

If you generically include programs in the CCMACINC table, you may need to add entries to the Exclude Table to prevent certain programs from being translated to command-level. If generic include statements could possibly include any of the following programs from the listed products, you must exclude them by adding entries to the Exclude table.

| | |
|---|---|
| Abend-aid | CCAA*, CCAS*, CCAT*, and WPIDRVC |
| ACF2 | ACFA* and ACF6* |
| CAFC | AFC* |
| CICS Comet | CC$$*, IVPA$G*, IVPC$G*, and IVPP$G* |
| CICS PLAYBACK | PLA* |
| CICS System programs | DFH*, DSNC*, and DSN2* |
| DADS | DADE*, DADR*, DADS*, DADM*, DADX*, and DPLUSOPT |
| EYEWITNESS | DASE* and DAST* |
| IMS | DFS* |
| InterTest | INT25* |
| Omegamon | OMOC* and EISET |
| RADAR | CWCS* and CWCT* |
| SYSD | SYSD* |
| XPEDITER | XPDBUG* |

For example, if you add a generic include INT* entry to the Include table, you must add an INT25* entry to the Exclude table to prevent the InterTest program from being translated. Some default exclude entries are part of the CCMACEXC sample member located in the SAMPLIB dataset.

## Example Entries within the CCMACEXC Exclude Table

The following example shows an excerpt from the CCMACEXC member located in the SAMPLIB dataset. Programs listed within CCMACEXC are excluded from translation to command-level by CICS Comet. Enter the program name beginning at column 1. You can use wildcards to generically exclude a range of programs with a common name segment.

```
/* -------- IVP MAPSETS DEFINED AS ASM -------------------------*/
IVPA$G*
IVPC$G*
IVPP$G*
/* */
/* -------- CICS SYSTEM PROGRAM ENTRIES ------------------------*/
DFH*                        EXCLUDE ALL CICS PROGRAMS
/* */
/* -------- IMS SYSTEM PROGRAM ENTRIES -------------------------*/
DFS*                        EXCLUDE ALL IMS PROGRAMS
/* */
/* -------- CICS SYSTEM PROGRAM ENTRIES ------------------------*/
DSNC*                       EXCLUDE ALL DB2 PROGRAMS
DSN2*                       EXCLUDE ALL DB2 PROGRAMS CICS640 AND UP
```

**Step 8: Additional PPT Entries**

Dynamic Attach ignores assembler programs defined by macro PPT entries; entries that were not created from your DFHCSD dataset. This means that assembler programs defined by macro entries in an assembled PPT table are not dynamically attached for macro-level conversion. This is not a consideration in CICS/TS because all PPT entries in CICS/TS are created through the CSD.


WARNING

In CICS/TS or CICS/TS, autoinstall cannot be used for application programs that run under the control of CICS Comet.

## Step 9: **Starting** CICS Comet

CICS Comet can be started in the PLT or with a command entered from a CICS 3270 terminal. To start CICS Comet from a terminal, enter the `HALLEY START` command. Restart CICS to start CICS Comet from the PLT.

1. Cycle your CICS region.
2. Start CICS Comet by entering the `HALLEY START` command from a terminal or by recycling CICS.

   ☞ CICS Comet must be started before certain applications. If you are starting Comet from the terminal, ensure these applications have not yet been started. See "PLT entry required for CICS Comet" on page 12, for a list of these applications.
3. Verify that CICS Comet is installed by entering the command HALLEY COMET from any CICS terminal.

Several messages appear during start-up to indicate that CICS Comet started successfully.

In CICS/TS and above, the CSV001 message indicates a real error. CICS Comet then issues a message indicating how to correct the error. Refer to "Messages and Codes" on page 83 for a listing of all CICS Comet messages.

**Step 10: Verifying Installation (Optional)**

This steps explains how to use the Installation Verification Program to verify that CICS Comet has been installed correctly.

The PPT and PCT entries necessary for the IVP can conflict with currently installed entries. You should install IVP only in a technical support region.

1. Run the IVPBUILD job located in the CNTL dataset.

   IVPBUILD allocates a CICS Comet IVP VSAM dataset. IVPBUILD deletes and defines a small VSAM KSDS dataset about 1 cylinder in length and loads 80 byte input records with 6 byte keys.

2. Add IVP PPT and PCT entries by using upgrade JCL from *hlq*.CCV540.SAMPLIB(COMETIVP).

   CICS/TS users only have to change the dataset name in the FCT entry for IVPFILEA to match the DSN created by the IVPBUILD process:

   *hlq*.CCV540.IVPFILEA.CL.
   The record format of the FCT entry must be VSAM fixed block.

   ☞ Existing users: These definitions have changed. You must perform this step to run the IVP.

3. Edit your CICS start-up JCL, and add the following DD statement to your DFHRPL concatenation:

   ```
   // DD DISP=SHR,DSN=hlq.CCV540.OPTIONAL
   ```

4. Add three entries to your macro Include table, CCMACINC, for the programs IVPC$MNU, IVPC$ALL, and IVPC$BRW with both flags set to blanks. Example:

   ```
   IVPC$MNU| | | | | | | | |  CICS Comet IVP
   IVPC$ALL| | | | | | | | |  CICS Comet IVP
   IVPC$BRW| | | | | | | | |  CICS Comet IVP
   ```

5. Cycle CICS to put these changes into effect.
6. Enter a $MNU or EMNU tranid and press ENTER.

   Tranids starting with $ are VS/COBOL macro-level programs running with Dynamic Attach. Tranids starting with E are relinked VS/COBOL macro-level programs running in 31-bit mode.

   A screen appears to enter a 4-byte tranid and a 6-byte record key. Type $BRW or EBRW in the tranid field and 000000 in the record key field, and press ENTER.

   This will display a screen with four records displayed on it. You can enter an F or a B in row 1 column 1 and press ENTER to go forward or backward through the IVPFILEA file.

To receive a detailed display of a record, select the key number from the browse and press CLEAR to end the browse function. Then, return to the $MNU or EMNU menu screen and type $INQ or EINQ in the tranid field and the key number you selected in the key number field, and press ENTER. You will receive a detailed display of the selected record.

☞ All the above transactions can be run using CEDF if you want to see what is going on behind the scenes.

CICS Comet transactions beginning with E are 31-bit macro-level. Do not test these transactions when a debugger is active.

# Chapter 3 Customization

This chapter describes optional Comet customization procedures. Some of these procedures may need to be completed to adapt CICS Comet to your site's system environment. Review each section in this chapter to determine if these procedures need to be completed at your site.

The following list shows the major customization tasks described in this chapter:


- Customize other software for CICS Comet (page 28)
- Customize the CCSIPARM member (page 30)
- Move BMS maps above the 16 Mb line (page 37)
- Run macro level programs above the 16 Mb line (page 38)
- Add EXEC CICS ADDRESS program entries to Include and Exclude tables (page 39)

### Customizing Other Software for Use with CICS Comet

This section discusses any considerations you must make prior to running CICS Comet with other software.

#### InterTest, XPEDITER, or ADSP Debuggers

The CICS Comet stub is added to beginning of assembler and VS/COBOL load modules. When you are examining the execution of these programs with a debugger, program offsets are not computed correctly by debuggers because of the length of the CICS Comet stub added to the beginning of the load module. PTFs are available from the vendors of InterTest, XPEDITER, and ADSP debuggers that allow them to work with CICS Comet. Refer to "CICS Comet Program Stub" on page 117 for a listing of the length of the stub added to COBOL, assembler, and PL/I programs.

Any execution debugging product must exclude CICS Comet from being monitored, just as CICS Comet excludes the execution debuggers from its macro level to command level conversion. The program prefixes to exclude CICS Comet from XPEDITER, InterTest, etc., are CC$$* and S2K* if running SAS System 2000 (S2K) database.

CICS Comet also supplies two optional InterTest user exits that allow macro level applications access to the CICS Comet CSA/CWA without InterTest trapping and producing a breakpoint. These exits are also provided in load module form in the CICS Comet load library with the prefix of IN*. The load module form of the exits must be renamed to the names used by InterTest to activate the exits.

- IN25UEXI: No rename is necessary.
- IN42LETX: Rename to IN25LETX for InterTest 4.2 and below.
- IN51LETX: Rename to IN25LETX for InterTest 5.1 and above.

After renaming the exits, make sure the CICS Comet load library precedes the InterTest load library in the CICS DFHRPL concatenation.

#### PL/I

The IBM PL/I "ON ERROR" handler must be relinked to change the residency mode to non-XA. This module is only 500 bytes long so it would be advisable to make it resident in the PPT by coding `RES=YES`. Sample JCL to relink the PL/I error handler is located in the IBMBEOCA member of the SAMPLIB dataset.

```
//JOB@NAME JOB     1,IBMBEOCA,CLASS=A,MSGCLASS=X,REGION=2048K
//LINKEDIT  EXEC   PGM=IEWL,PARM='XREF,LIST,LET,AMODE(31),RMODE(24)'
//SYSLIB    DD     DISP=SHR,DSN=PLI.R23.SIBMLINK  ORIGINAL LOADLIB
//SYSPRINT DD      SYSOUT=*
//SYSUT1    DD     UNIT=VIO,SPACE=(1024,(100,50))
//SYSLMOD DD       DISP=SHR,DSN=PLI.R23.PLILINK(IBMBEOCA)  NEW LOADLIB
//SYSLIN    DD *
  INCLUDE SYSLIB(IBMBEOCA)
//
```

# MSA

Perform the procedures in “Supporting Direct CICS Control Block Search”, “Coding Non-XA FCT entries” on page 70. Use member DFHFCTMD of the SAMPLIB as your base, rather than member DFHFCTCC.

DFHFCTMD contains the entries used by MSA and its Information Expert product.

# Notis

If you are running Notis 501, please review member $NOTIS of the SAMPLIB dataset. Implementing the procedures in this member will allow you to run Notis under Comet.

# Customizing CCSIPARM

The CCSIPARM member contains parameters that modify CICS Comet default operating characteristics. CCSIPARM is located in the CCV540.SAMPLIB dataset.

The following list shows the possible values that can be assigned to CCSIPARM parameters. Default values are underlined. Values that you can assign to a parameter are *italicized*.

| | |
|---|---|
| DEST=CSMT | WRITE CICS/COMET MESSAGES TO "CSMT" LOG |
| /* ADDRESS_INCLUDES=CCCMDINC | *** REMOVED IN CICS/COMET 5.4.0 *** |
| /* ADDRESS_EXCLUDES=CCCMDEXC | *** REMOVED IN CICS/COMET 5.4.0 *** |
| ASM-RMODE(31)=NO | DISABLE COMET BI-MODAL ASM EXECUTION |
| /* BMS=YES | *** REMOVED IN CICS/COMET 5.4.0 *** |
| /* COBOL=YES | *** REMOVED IN CICS/COMET 5.4.0 *** |
| CSA=NO | *** NO IS NOW THE DEFAULT IN COMET 5.4.0 |
| CSA_RELEASE=NO | DETERMINE CSA RELEASE NUMBER IN COMET CSA |
| CWA=ONE | PLACE COMET CWA @ IN REAL CWA @ LOCATION |
| CWA_KEY=USER | COMET'S CWA COPY IS TO BE IN "USER" KEY |
| DB2=NO | ONLY NECESSARY WHEN NOT USING DYNAMIC ATT. |
| DFHCSAD=NO | ONLY NECESSARY WHEN NOT USING DYNAMIC ATT. |
| DYNAMIC=YES | USE CICS COMET'S DYNAMIC ATTACH |
| EXITS=NO | DISABLE USE OF CICS COMET EXITS |
| KEYS=ANY | ASSEMBLER FCP KEYS RMODE=ANY\|BELOW |
| /* LE=NO | *** REMOVED IN CICS/COMET 5.4.0 *** |
| LOG=NO | DISABLE MACRO STATEMENT LOGGING |
| MACRO_ASM_DOPE_VECTORS=YES | BUILD DOPE VECTORS FOR MACRO ASM PROGRAMS |
| OPEN=YES | SPECIFY HOW TO HANDLE FILE OPENS |
| PLI=YES | ENABLE COMET PL/I SUPPORT |
| PROGRAM_INCLUDES=CCMACINC | SPECIFY MEMBER NAME OF CCMACINC TABLE |
| PROGRAM_EXCLUDES=CCMACEXC | SPECIFY MEMBER NAME OF CCMACEXC TABLE |
| TRACE=YES | ENABLE MACRO STATEMENT TRACING |
| TRAN_INCLUDES=CCTRNINC | SPECIFY TRANAACTION NAME IN CCTRNINC TABLE |
| TRAN_EXCLUDES=CCTRNEXC | SPECIFY TRANSACTION NAME IN CCTRNEXC TABLE |
| DCT=NO | SET "END OF TABLE" INDICATOR IN CSADCTBA |
| FCT=YES | PLACE REAL FCT ADDRESS IN COMET CSAFCTBA |
| JCT=NO | SET "END OF TABLE" INDICATOR IN CSAJCTBA |
| PCT=NO | SET "END OF TABLE" INDICATOR IN CSAPCTBA |
| PPT=NO | SET "END OF TABLE" INDICATOR IN CSAPPTBA |
| TCT=YES | PLACE REAL TCT ADDRESS IN COMET CSATCTBA |
| TST=NO | SET "END OF TABLE" INDICATOR IN CSATSTBA |
| CUSTOMER_NAME=*name* | PASSWORD PARAMETER: DO NOT CHANGE |
| CPUID=*cpuid* | PASSWORD PARAMETER: DO NOT CHANGE |
| MODEL=*model* | PASSWORD PARAMETER: DO NOT CHANGE |
| EXPIRATION_DATE=*yyddd* | PASSWORD PARAMETER: DO NOT CHANGE |
| PASSWORD=*password* | PASSWORD PARAMETER: DO NOT CHANGE |
| PRODUCT_CODE=*code* | PASSWORD PARAMETER: DO NOT CHANGE |

You should not change CCSIPARM parameter values. Instead, create a new member and include only those parameters you want to override (similar to CICS SIP overrides). Name your override members with either of the following formats:

| | |
|---|---|
| applid or jobname | applid of the region's VTAM application ID |
| jobname | APPLID will be ingnord if it is the same name as jobname. |

Note to CICS/COMET 3.3.0 and lower users: CC$$SYSID is no longer supported in version CICS/COMET 3.3.1 and above.

Comet looks for CCSIPARM first when it initializes. CCSIPARM must be present or CICS Comet initialization terminates. Then, CICS Comet looks for the applid member. If applid is not found, it looks for the jobname member. If either member is found, CICS Comet uses their parameter values to override the CCSIPARM defaults.

**CCSIPARM Parameters**

You may want to override the following parameters:

- ADDRESS_INCLUDES and ADDRESS_EXCLUDES
- LOG
- OPEN
- PROGRAM_INCLUDES and PROGRAM_EXCLUDES
- TRACE

Otherwise, use the supplied CCSIPARM defaults unless an UNICOM Systems Customer Service representative tells you to change them.

ADDRESS_INCLUDES and ADDRESS_EXCLUDES=*member*

Member names of Command Level Include and Exclude tables. You can specify up to four member names with each parameter. This allows you to place Include and Exclude statements into functional groups. For example, you could code:

```
ADDRESS_INCLUDES=AR,AP,PAYROLL,SHIPPING
```

The defaults, CCCMDINC and CCCMDEXC, are automatically appended to the end of each list. (These members are in addition to the four maximum you can specify.) Default members are processed last, even if they are included earlier in the list. Otherwise, they are processed in the order they are listed.

ASM-RMODE(31)=[NO|YES]

Choice (YES or NO) to enable the Dynamic Attach facility to move assembler programs above the 16 Mb line. NO is the default.

BMS=[YES|NO] Choice (YES or NO) to allow CICS Comet to place BMS maps above the 16 Mb line. For all CICS/TS releases, CICS Comet places only mapsets above the 16 Mb line that are defined in the PPT as USAGE=MAP. YES is the default.

CC$E_TRAP=[YES|NO]

Choice (YES or NO) to trap and abend application programs with a “DFHE” stub that appear to be macro-level programs. YES is the default.

CC$X_TRAP=[YES|NO]

Choice (YES or NO) to trap and abend application programs without command-level stubs that appear to be command-level programs. YES is the default.

CC$Y_TRAP=[YES|NO]

Choice (YES or NO) to trap and abend application programs with “DFHY” stubs that appear to be macro-level programs. YES is the default.

CHECK=[YES|NO] Choice (YES or NO) to check the TWA for storage overlays each time a macro-level request is issued by the application program. Overlays result when an application program uses more TWA storage than specified in the CICS PCT entry for a transaction. YES is the default.

COBOL=[YES|NO] Choice (YES or NO) to enable CICS Comet to perform a COBOL/VS bi-modal self-test. COBOL=YES is required if you have VS/COBOL I programs compiled with the RES option, or if you are using restricted COBOL VERBS such as UNSTRING or INSPECT. YES is the default.

Set the parameter to COBOL=NO if you do not have any VS/COBOL programs, or only have COBOL II, PL/I, or ASM programs.

COBOL2=[YES|NO] Obsolete Cobol Runtime parameter, last used on CICS release TS 2.2, which is no longer supported by IBM or Unicom.

COMET_TO_SATISFY_1ST_RECEIVE=[YES|NO]

Choice (YES or NO) to allow Comet to satisfy the command-level application program's first "EXEC CICS RECEIVE" request with data received by Comet under the control of the CICS Execution Diagnostic Facility (EDF). YES is the default.

CSA=[AUTO|NO|YES]

AUTO equates to YES under CICS 4.1 and below. The command tables are used. NO turns off CICS Comet's EXEC CICS ADDRESS CSA/CWA/TWA/etc. substitution logic. Refer to "Command Level Programs—CSA ADDRESS" on page 39 for more information about this option.

CSA_RELEASE=[NO|*nn*]

2-digit number of the CICS version that works with CICS Comet. Valid values are 21, 31, 32, 33, 41, 51, 52, 53, 61, 62, 63, and 65. If NO, the CICS Comet CSA specifies the release of CICS actually in use. NO is the default.

CWA=[ONE|NO|YES]

Choice to store the CICS Comet CWA address in the CICS CSA in place of the real CICS CWA (This feature only applies to all CICS/TS releases, and is the same as specifying NO in prior CICS releases).

CWA_KEY=[USER|CICS]

Storage key to acquire memory for Comet's copy of the CWA. The CWA copy is always placed below the 16 Mb line at either the UDSA or the CDSA depending upon the CWA_KEY value. (Do not change this option unless advised to do so.) USER is the default.

CWA_OFFSET=[00512|nnnnn]

Byte offset from the start of the CSA to the beginning of the Comet CWA. The default is 512 bytes or X'200'.

DB2=[NO|YES] Choice (YES or NO) to allow CICS Comet to filter DB2 and DATACOM/DB requests to determine whether it should process them. Filtering is necessary for mixed-mode processing. This is only necessary when you are not using Dynamic Attach for all programs. Otherwise, this option is set for each program using CCMACINC flags. NO is the default.

DEST=[CSMT|*xxxx*]

DCT destination for CICS Comet start-up messages. CSMT is the default.

DFHCSAD=[NO|YES]

Choice (YES or NO) to allow CICS Comet to replace DFHCSAD TYPE=LOCATE macros with the CICS Comet equivalent. Replacement is necessary only if Dynamic Attach for all programs is not being used. Otherwise, this option is set for each program using CCMACINC flags. NO is the default.

DYNAMIC=[YES|NO]

Choice (YES or NO) to use the Dynamic Attach facility to add the CICS Comet stub to macro level programs. Dynamic Attach makes relinking macro level application programs unnecessary. YES is the default.

In CICS/TS, BMS mapsets can be placed above the 16M line regardless of whether Dynamic Attach is used or not.

EXEC=[YES|NO] Choice (YES or NO) to monitor dynamically attached command-level programs that execute “EXEC CICS” commands. The default is YES.

EXITS=[NO|YES] Choice (YES or NO) to enable CICS Comet exits. If you set EXITS to NO, no exits will be executed, whether they are customized or not. NO is the default.

KEYS=[ANY|NO] Choice to support 31-bit keys used in file control accesses above the 16MB line. ANY is the default.
Normally, file control access keys are specified in the KEYS or RIDFLD field of macro-level applications. The fields reside in the following storage areas based upon the language the program was written:

| | |
|---|---|
| Assembler | TWA or GETMAIN area |
| COBOL | Working storage |
| PL/I | Dynamic Save Area (DSA) |

These storage areas reside below the 16MB line and present no problem to 31-bit application programs running under the control of CICS Comet. Occasionally, the keys are placed in static or program storage, which can cause problems. KEYS=ANY becomes necessary when RMODE=Y is used in CCMACDNC. CICS Comet moves the keys above the 16MB line along with the program. KEYS=NO reduces CPU usage slightly.

LE=[NO|YES|AUTO] ☞ *This option is now Obsolete, as Auto is the default.*

Choice to use the Language Environment (LE) interface for programs running under CICS Comet control.

| | |
|---|---|
| NO | Programs running under CICS Comet control do not use the Language Environment interface. PL/I programs require the PL/I run-time environment. AUTO is the default. |
| YES | Programs running under CICS Comet control use the Language Environment interface. The Language Environment for CICS interface must be active. The CICS release must be 4.1 and above, and you must have a permanent CICS Comet password. |
| AUTO | Let Comet figure out which library you are using in steplib or linklist. |

LOG=[NO|YES|*suffix*]

Choice to write an entry in the macro statement log each time CICS Comet encounters a macro-level statement. NO is the default.

The entry shows the following information (on a single line):

```
DFHTR MACRO-LEVEL CODE FOUND IN PROGRAM: IVPC$MNU  TRAN=$MNU  2005.261  08:59:04
```

The macro statement log is stored in the dataset defined by the

COMETMAC DD statement. The EXEC CICS ADDRESS CSA log is kept in the dataset defined by the COMETCSA DD statement for CICS TS.

LOG=YES imposes a substantial performance overhead on your system. Use LOG=YES only for debugging or testing purposes.

MACRO_ASM_DOPE_VECTORS=[NO|YES]

Choice (YES or NO) to have CICS Comet build dope vectors for the Execute Interface Block (EIB) and COMMAREA addresses.

Some macro-level programs were written with the knowledge that CICS presented the EIB and COMMAREA addresses with dope vectors. PL/I uses dope vectors; COBOL and assembler programs do not. Command-level strips the dope vectors for COBOL and assembler programs when the EXEC CICS environment is built for the receiving program. However, if the applications getting linked to were not command-level before CICS Comet, they expect to see and then strip the dope vectors off themselves before using the EIB and COMMAREA addresss. NO is the default.

OPEN=[YES|NO|INITIAL]

Choice to open files in the pseudo FCT at CICS Comet initialization. Files are opened and queried to add data (such as record size and key length) to the pseudo FCT. YES is the default.

If YES, all files are opened and queried regardless of the FILSTAT setting. If INITIAL, only files with FILSTAT=OPEN are opened and queried. With INITIAL, you must put CC$$SOCC in the CICS/TS PLTPI to start the CC$$XOCC user exit. If NO, no files are opened at CICS Comet initialization.

☞ Do not code NO unless you are instructed to do so by a UNICOM Systems Customer Service representative.

PLI=[YES|NO] CHOICE (YES or NO) to provide CICS Comet support for PL/I programs running in CICS/TS regions. YES is the default.

PROGRAM_INCLUDES and PROGRAM_EXCLUDES=*member*

Member names of Macro Include and Exclude tables. You can specify up to eight member names following both parameters. This allows you to break your Include and Exclude statements into functional groups. For example, you can code:

```
PROGRAM_INCLUDES=AR,AP,PAYROLL,SHIPPING
```

The default members, CCMACINC and CCMACEXC, are appended automatically to the end of each parameter list. These members are in addition to the eight maximum that can be specified. The default members are always processed last, even if they are included earlier in the list. Otherwise, all other members are processed in the order they are listed with both parameters.

TRACE=[YES|NO] Choice (YES or NO) to trace every occurrence of macro-level statements encountered by CICS Comet within a program. Trace entries are written to the CICS internal trace table. YES is the default.

In CICS/TS, the trace entry contains the contents of the CICS Comet TCA and the application return address (R14).

Use TRACE=YES only for debugging purposes. Tracing must be active to debug SVC or transaction dumps. Although CICS/TS tracing uses relatively little CPU resources, you should set `TRACE=NO` in a stable production environment, or before running a benchmark test.

DCT, FCT, JCT, PCT, PPT, TCT, and TST=[NO|YES|*suffix*]

If specified as NO, such as PPT=NO or PCT=NO, CICS Comet places a hexadecimal fullword of FFFFFFFF in the CICS Comet CSA to indicate the end of the CICS table.

These tables are placed above the 16 Mb line for CICS/TS. You must specify NO, or use a 2-character suffix that identifies the table you assembled by completing the procedures described in “Supporting Direct CICS Control Block Search” on page 70. YES defaults to NO under CICS/TS.

TS_QUEUED=[NO|YES]

Choice (YES or NO) to allow CICS Comet to substitute DFHTS GET/PUT macros with GETQ/PUTQ.

The default is YES for CICS/TS because single TS QUEUEs are no longer supported by the CICS Command Level interface.

**Password Parameters**

Do not change any of the following parameter values except under the guidance of a UNICOM Systems Customer Service representative. These parameters affect how CICS Comet passwords are processed.

! WARNING

Password parameters must match similar password information given to you by UNICOM Systems. Changing any of the following Comet parameters invalidates your password and disables CICS Comet..

| | |
|---|---|
| CUSTOMER_NAME=name | Company name, as specified by UNICOM Syste ms enclosed within quotes. This field is exactly 40 characters long. Any unused portion of the field is padded with blanks up to the 40 character limit. |
| CPUID=cpuid | CPU ID. |
| MODEL=model | CPU model number. |
| EXPIRATION_DATE=yyddd | CICS Comet Julian expiration date. |
| PASSWORD=password | New CICS Comet password. |
| PRODUCT_CODE=code | CICS Comet product code. |

# Moving BMS Maps Above the 16 Mb Line

All CICS/TS releases allow BMS maps to be relinked with `AMODE=31` and `RMODE=ANY` to place them above the 16 Mb line. Earlier versions of CICS do not support BMS maps above the line. With CICS Comet, BMS maps can be placed above the 16 Mb in any CICS TS region.

You can specifically relink BMS maps with `AMODE=31` and `RMODE=ANY`. Or, you can allow CICS Comet to automatically move maps above the 16 Mb line.

| | |
|---|---|
| CICS version 2.1 | CICS Comet moves all maps above the 16 Mb line. |
| All CICS/TS releases | CICS Comet's Dynamic Attach dynamically moves all BMS maps defined as "MAPSETS" in the PPT to the extended DSA. |

Note: The CICS Comet system-wide statistics screen shows the amount of virtual storage saved by having CICS Comet move the BMS Mapsets above the 16 Mb line on CICS TS. Enter the HALLEY (blank) command from any CICS terminal to display the system-wide statistics screen. An example of the screen is shown on page 115.

**Sample Relink for BMS Mapsets - SAMPLIB(BMSMAPS)**

```
//JOB@NAME JOB 1,BMSMAPS,CLASS=A,MSGCLASS=X,REGION=2048K
//*-------------------------------------------------------------------
//*      SAMPLE RELINK OF AN BMS MAPSETS FOR CICS COMET BMS XA
//*-------------------------------------------------------------------
//LINKEDIT EXEC      PGM=IEWL,PARM='REUS,XREF,LIST,LET,AMODE(31),RMODE(ANY)'
//SYSLIB             DD DISP=SHR,DSN=your.old.non.xa.BMS.loadlib
//                   DD DISP=SHR,DSN=&INDEX.CCV540.LOAD
//SYSUT1             DD UNIT=VIO,SPACE=(1024,(100,50))
//SYSPRINT           DD  SYSOUT=*
//SYSLMOD            DD DISP=SHR,DSN=new.xa.BMS.loadlib(mapset)
//SYSLIN             DD *
    INCLUDE SYSLIB(mapset)
/*
//
```

# Run Macro Level Programs Above the 16 Mb Line

CICS Comet macro application programs can run in XA mode (above the 16 Mb line) offering significant virtual storage constraint relief by specifying `AMODE=31` and `RMODE=ANY` on the link edit PARM statement.

In CICS/TS, the Dynamic Attach Include table, CCMACINC, includes an RMODE column where you can specify Y for Yes to allow CICS Comet to move and execute a program above the 16 Mb line. You can also relink your programs and change the `RMODE` parameter. See the JCL examples in "Examples of Relink JCL" on page 59. RMODE is ignored in CICS 2.1.

Restriction: CICS Comet removes the terminal attached indicator from the TCA field TCAFCAAA. Programs that test for this indicator must remain in 24-bit mode.

☞ Test all programs under CICS Comet below the 16 Mb line (24 bit mode) before you test them above the 16 Mb line (31 bit mode). Migrate programs from 24 bit mode to 31 bit mode one at a time to identify any programs that cannot operate above the 16 Mb line.

Refer to the CCSIPARM KEYS parameter on page 33 for information about moving 31-bit file control access keys above the 16-Mb line.

# Command Level Programs—CSA ADDRESS

The sample Include and Exclude tables distributed with CICS Comet have entries for command level programs. These command level entries are suitable for most installations and probably do not need to be changed. However, you may need to add entries to the Include table for third party software products that require the real CSA address.

EXEC CICS ADDRESS CSA is no longer supported in CICS version 3.2 and above. However, CICS Comet can present the CSA it builds to command level programs that issue EXEC CICS ADDRESS CSA calls. This can be very useful for application systems that use both command level and macro level programs. The only requirement is that you use the CICS 2.1 translator on all command level programs that contain the EXEC CICS ADDRESS CSA calls.

CICS Comet always presents its CSA to all macro level programs under its control.

The CCCMDINC Include and CCCMDEXC Exclude table work together to specify which programs receive the CICS Comet CSA, CWA, and TWA versus the real CICS CSA, CWA, and TWA.

The rules for specifying programs in the CCCMDINC and CCCMDEXC tables are the same as for the CCMACINC and CCMACEXC tables. See “Step 7: Dynamic Attach for Macro-Level Translation,” on page 17, for more information.

## Listing Included and Excluded Programs

When CICS Comet is active, you can display the list of included command level programs by entering `HALLEY LIST INCLUDED COMMAND PROGRAMS` or `HALLEY L I C` from any CICS terminal. To display the command level programs that have been excluded, enter `HALLEY LIST EXCLUDED COMMAND PROGRAMS` or `HALLEY L E C`.

Both commands list the programs in the order they are entered in the table. If the list exceeds one page, “MORE…” appears at the bottom right corner of the screen. Press ENTER to display the next page of programs.

## Making Dynamic Changes to Tables

You may change the Include/Exclude tables at any time using ISPF/PDF, but you must enter HALLEY REFRESH COMMAND from any CICS terminal to refresh the CICS Comet tables in memory.

## Different Tables for Different Regions

You can have different Include and Exclude tables for different CICS regions. You can assign any name to the tables, However, all tables must be stored in the Comet SAMPLIB dataset specified in your CICS start-up JCL.

To have different tables for different regions, perform the following:

1. Set up different CCSIPARM members for different regions. See “Customizing CCSIPARM” on page 30, for details.

2. In the CCSIPARM member controlling that region, specify the Include and Exclude table names with the ADDRESS_INCLUDES and ADDRESS_EXCLUDES parameters.

**CCCMDINC Include Table**

The following example is an excerpt from the CCCMDINC member located in the SAMPLIB dataset used by CICS Comet. CCCMDINC includes command level programs to replace EXEC CICS ADDRESS CSA requests.

```
/*------|-|-|-|-|-|-|-|-|---------------------------------------
/*      |C|C|T|-|-|-|-|-|
/*      |S|W|W|-|-|-|-|-|
/*      |A|A|A|-|-|-|-|-|
/* PROG |-|-|-|-|-|-|-|-|
/* NAME |@|@|@|-|-|-|-|-|  PROGRAM DESCRIPTION...
/*------|-|-|-|-|-|-|-|-|---------------------------------------
DFHECID | | | | | | | | | SPECIFIC INCLUDE FOR EXEC CICS
/*    | | | | | | | | | | INTERACTIVE
DFHECIP | | | | | | | | | SPECIFIC INCLUDE FOR EXEC CICS
/*    | | | | | | | | | | INTERACTIVE
******* | | | | | | | | | INCLUDE ALL PROGRAMS FOR EXEC CICS
/*      | | | | | | | | | ADDRESS CSA
```

Begin each entry in column 1. For each entry in the Include table, you can specify:

| | |
|---|---|
| CSA | Y for Yes in the CSA column designates the CICS Comet copy of the CSA should be used for this program. N for No designates the real CSA should be used. |
| CWA | Y for Yes in the CWA column designates the CICS Comet copy of the CWA should be used for this program. N for No designates the real CWA should be used. |
| TWA | Y for Yes in the TWA column designates the CICS Comet copy of the TWA should be used for this program. N for No designates the real TWA should be used. Native Command level programs always receive the real TWA. Macro level programs under CICS Comet's control always receive the CICS Comet copy of the TWA.<br><br>Some mixed-mode programs may need the real TWA. If your mixed-mode program abends, specify N for No to use the real TWA. |

☞ If all three fields are left blank, CICS Comet defaults to Y for Yes for all columns. The defaults are appropriate for most command level programs.

## Example

```
/*      |C|C|T|-|-|-|-|-|
/*      |S|W|W|-|-|-|-|-|
/*      |A|A|A|-|-|-|-|-|
/* PROG |-|-|-|-|-|-|-|-|
/* NAME |@|@|@|-|-|-|-|-|  PROGRAM DESCRIPTION...
/*------|-|-|-|-|-|-|-|-|---------------------------------------
SAMPLE1 | | | | | | | | |  Use Comet copies of CSA, CWA, & TWA.
SAMPLE2 |Y|Y|Y| | | | | |  Use Comet copies of CSA, CWA, & TWA.
SAMPLE3 |Y|Y| | | | | | |  Use Comet copies of CSA and CWA,
/*                          and use real TWA.
```

**CCCMDEXC Exclude Table**

The following example shows the CCCMDEXC member in the SAMPLIB dataset. CCCMDEXC contains entries to exclude command level programs from CSA ADDRESS substitution. If you do not have any programs that are in the Include table (CCCMDINC), then exclude processing is meaningless.

Begin each entry in column 1.

```
/* ----------- CICS SYSTEM PROGRAM ENTRIES ----------------------*/
DFH*                    EXCLUDE ALL CICS PROGRAMS
/* ----------- CICS/MANAGER ENTRIES -----------------------------*/
CMR*                    CICS/MANAGER FROM BOOLE AND BABBAGE
/* ----------- INTERTEST ENTRIES --------------------------------*/
INT25*                  INTERTEST FROM ON-LINE SOFTWARE INT'L
/* ----------- CICS/OMEGAMON ENTRIES ----------------------------*/
OMOC*                   OMEGAMON/CICS FROM CANDLE
EISET                   OMEGAMON/CICS FROM CANDLE
/* ----------- ACF2 ENTRIES -------------------------------------*/
ACFA*                   ACF/2 MASTER TRANSACTION
ACF6*                   ACF/2 MASTER TRANSACTION
/* ----------- DADS ENTRIES -------------------------------------*/
DADE*                   DADS FROM ON-LINE SOFTWARE INT'L
DADR*                   DADS FROM ON-LINE SOFTWARE INT'L
DADS*                   DADS FROM ON-LINE SOFTWARE INT'L
DADM*                   DADS FROM ON-LINE SOFTWARE INT'L
DADX*                   DADS FROM ON-LINE SOFTWARE INT'L
DPLUSOPT                DADS FROM ON-LINE SOFTWARE INT'L
/* ----------- CAFC ENTRIES -------------------------------------*/
AFC*            CAFC FROM NETEC INTERNATIONAL INC.
/* ----------- EYEWITNESS ENTRIES-------------------------------*/
DASE*                   EYEWITNESS FROM LANDMARK
DAST*                   EYEWITNESS FROM LANDMARK
/* ----------- RADAR ENTRIES ------------------------------------*/
CWCS*                   CICS RADAR FROM COMPUWARE
CWCT*                   CICS RADAR FROM COMPUWARE
/* ----------- PLAY-BACK ENTRIES --------------------------------*/
PLA*                    CICS PLAYBACK
/* ----------- DBUG-AID / XPED ENTRIES --------------------------*/
DBUG*                   CICS RADAR FROM COMPUWARE
/* ----------- ABEND AID ENTRIES --------------------------------*/
CCAA*                   CICS ABEND AID FROM COMPUWARE
CCAS*                   CICS ABEND AID FROM COMPUWARE
CCAT*                   CICS ABEND AID FROM COMPUWARE
WPIDRVC                 CICS ABEND AID FROM COMPUWARE
/* ----------- SYSD ENTRIES -------------------------------------*/
SYSD*                   SYSD FROM H&W SOFTWARE
```

# Chapter 4 Debugging CICS Comet

CICS Comet has been tested in many environments. If you are experiencing a problem, review the following trouble shooting suggestions to rule out common errors before contacting UNICOM Systems Customer Service.

- Verify that all system software products (Omegamon, DADS, TMON, OmniGuard, etc.) have been added to both the macro and command level Exclude tables.
- A common error is to include or relink only the high-level or first program in a transaction and neglect to include or relink other programs that are LINKed or XCTLed to from the high-level program. Make sure that you have included or relinked all programs that are executed by one transaction. This will be evident if you watch the macro programs run under CEDF.
- If you have a macro level program that does not operate correctly with CICS Comet, it may be using non-standard CICS facilities that are not apparent. CICS Comet has been designed to handle many non-standard macro programs. The most frequent error message seen from macro level programs is “FCT or file not found.” If you receive a message similar to this, read the sections in this manual on creating entries for macro level programs that need to index or directly access the FCT, DCT, PPT, PLT, or TCT tables. (See “Supporting Direct CICS Control Block Search”, beginning on page 70.)
- If you receive ABM0 or ABMx abends from your application programs, verify that your BMS maps are defined as “MAPSETS” in the PPT or through CEDA. This is because maps defined as assembler are treated by Dynamic Attach as assembler programs and have the CICS Comet command level stub prefixed to them if you generically include this program name in CCMACINC. You could also exclude the BMS map in CCMACEXC if you included it generically in CCMACINC.
- If you receive abend AEXS or AEXT from macro level programs, follow the instructions in the section on relinking macro level DL/I programs that use static qualified Segment Search Arguments.
- If you have reinstalled CICS Comet or upgraded to a new release, verify that you are using the correct release of CICS Comet in your CICS system by issuing the HALLEY VERSION command.
- Turn on CEDF and monitor the macro level program that is not executing correctly. This will help you determine what problems the macro level program is having in execution. Print the ten previous CEDF displays prior to encountering the problem and have them available when you contact CICS Comet technical support.

If your problem is not resolved by any of these suggestions, collect the following information prior to contacting UNICOM Systems Customer Service.

- Copy of the screen display from the HALLEY VERSION command.
- Copies of any CICS Comet messages that were issued when the problem occurred.
- Access to any dumps produced when the problem occurred. Obtain an IBM CICS transaction dump and make sure the CICS in-core trace and Comet's trace are active. Pre-analyzed dumps produced by Abend-aid or EYEWITNESS are insufficient to resolve most problems. However, you can export RADAR dumps into SVC format.
- If the problem occurs when you are monitoring the macro level program with CEDF, then screen print all related CEDF displays before (a minimum of ten) and after the problem occurs. Include the PSW and registers (PF12 from the CEDF screen).
- Compiled source listing of the macro level program with the expanded assembler listing if the program is written in COBOL or PL/1.
- Enter HALLEY SNAPDUMP from any CICS terminal to create a snap dump on the CICS' dump dataset, if you do not send an SVC dump.
- Complete the information sheet on page 52, regarding program abends.

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

Voice 818-838-0606 Fax 818-838-0776
Website http://www.unicomsi.com/support
Email support@unicomsi.com

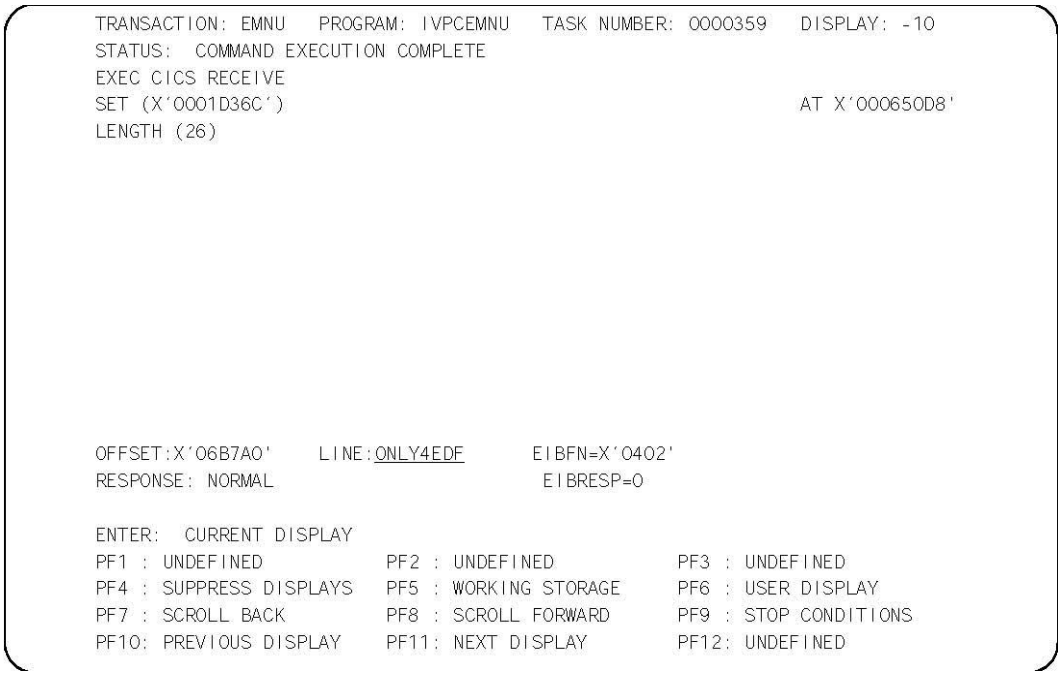
```
HALLEY VERSION
================================================================================
Cics Comet V5R1M0 COPYRIGHT(C): 1991-->1999,2000-->2011 (DR.CICS) 08/23/11 14.46
================================================================================
CICS COMET CURRENT DATE.......: 2011/12/14
CICS COMET EXPIRATION DATE....: PERM
CICS COMET Licensed CPUid.....: 037B36,2098/J04
CICS COMET LICENSED PARMS.....: 07301954 Unicom Systems Intl, Inc.      ,
================================================================================
CC$$EBFP: V5R1M0 08/21/11 1F2DF000 62B0  CC$$EBMS: V5R1M0 08/21/11 1F2E5300 33F0
CC$$EDCP: V5R1M0 08/21/11 1F2E8700 3000  CC$$EDIP: V5R1M0 08/21/11 1F2EB700 3280
CC$$EDLI: V5R1M0 08/21/11 1F2EEA00 3270  CC$$EFCP: V5R1M0 08/21/11 1F2F1D00 5EF8
CC$$EICP: V5R1M0 08/21/11 1F2F7C00 3288  CC$$EJCP: V5R1M0 08/21/11 1F2FAF00 3000
CC$$EKCP: V5R1M0 08/21/11 1FB9D000 36C0  CC$$EPCP: V5R1M0 08/21/11 1FBA0700 5B20
CC$$ESCP: V5R1M0 08/21/11 1FBA6300 3000  CC$$ESPP: V5R1M0 08/21/11 1FBA9300 3000
CC$$ETCP: V5R1M0 08/21/11 1FBAC300 3280  CC$$ETDP: V5R1M0 08/21/11 1FBAF600 39E0
CC$$ETRP: V5R1M0 08/21/11 1FBB3000 3000  CC$$ETSP: V5R1M0 08/21/11 1FBB6000 3598
CC67LDLD: V5R1M0 08/21/11 1FBBBF50 31A0  CC$$EXEC: V5R1M0 08/21/11 00244904 5930
CC$$CMCI: V5R1M0 08/21/11 00240000 5400  CC$$CPEI: V5R1M0 08/21/11 00233A30 0000
CC$$MAIN: V5R1M0 08/21/11 0022F228 54E0  CC$$KERN: V5R1M0 08/21/11 0022F228 5930
CC67EEIP: V5R1M0 08/21/11 002800B0 8430  CC67EPLI: V5R1M0 08/21/11 000BB7C8 3238
CC67EIPA: V5R1M0 08/21/11 1FBB96B0 1830  CC$$CCEE: V5R1M0 08/21/11 000B8560 3000
================================================================================
JobName=LJLDFH67 APPLID=LJLDFH67 SYSID=LL67 CICS_VERSION=67.12 RUNTIME_ENV=LE-34
```

☞ Please note that in 5.4.0, the version is 4 and the release is 2. The last digit of 0 is the maintenance level. Only domains that have been altered to support a new Version of CICS or have PTFs applied, will show an updated Modification level. For Example CC$$STRT might be at 4.2.1 and CC$$EFCP could be 5.4.0. This is intentional so that we can detect what maintenance level each module is at from a single screen.

# Using CEDF With CICS Comet

You can use the IBM CEDF debugging transaction on Cometized macro level programs to observe EXEC CICS commands being issued and their responses. Both the commands and the responses can be changed by overtyping them on the screen and pressing ENTER.

When an application program running under CICS Comet is run through CEDF, CICS Comet inserts an EXEC CICS RECEIVE into the initialization process to synchronize the input TIOA areas. Otherwise, the macro level application program would receive the CEDF TIOA area instead of the original TIOA area used to initiate the transaction. This is only done in the first program executed by the transaction. This EXEC CICS RECEIVE command can be identified by ONLY4EDF in the LINE field on the CEDF screen.



**Example of a Macro Level Request and Its Command Level Equivalent**

The following screens show a macro level request and the corresponding command level request generated by CICS Comet, as seen by CEDF.

## Macro Level Request

```
ST TIOABAR,TCTTEDA  tell BMS where the data is
DFHBMS TYPE=(OUT,ERASE,WAIT),MAP=MENU,MAPSET=DFHC$GA
```

“LINE: 826145AA” points to the first instruction after DFHBMS in this case. (See the CEDF screen below.)

☞ When running a macro program under CEDF, CICS Comet must insert an EXEC CICS RECEIVE command into the program so that the macro program sees the real input TIOA, not the last CEDF screen. It is identified by ONLY4EDF in the LINE # field.

## Command Level Equivalent under CEDF

```
TRANSACTION: EMNU   PROGRAM: DFHCEMNU   TASK NUMBER: 00141   DISPLAY:00
STATUS:  ABOUT TO EXECUTE COMMAND
EXEC CICS SEND MAP
MAP ('MENU    ')
FROM(`                                          `)
LENGTH (1920)
MAPSET ('DFHC$GA')
TERMINAL
ERASE









OFFSET:X'F04648'    LINE:826145AA       EIBFN=X'1804'


ENTER:  CONTINUE
PF1 : UNDEFINED            PF2 : SWITCH HEX/CHAR    PF3 : END EDF SESSION
PF4 : SUPPRESS DISPLAYS    PF5 : WORKING STORAGE    PF6 : USER DISPLAY
PF7 : SCROLL BACK          PF8 : SCROLL FORWARD     PF9 : STOP CONDITIONS
PF10: PREVIOUS DISPLAY     PF11: UNDEFINED          PF12: ABEND USER TASK
```

## Example of a Command Level Response

```
TRANSACTION: EMNU   PROGRAM: DFHCEMNU   TASK NUMBER: 00141   DISPLAY:00
STATUS:  COMMAND EXECUTION COMPLETE
EXEC CICS SEND MAP
MAP ('MENU    ')
FROM(`                                          `)
LENGTH (1920)
MAPSET ('DFHC$GA')
TERMINAL
ERASE









OFFSET:X'F04648'    LINE:826145AA      EIBFN=X'1804'
RESPONSE: NORMAL                       EIBRESP=0

ENTER:  CONTINUE
PF1 : UNDEFINED             PF2 : SWITCH HEX/CHAR     PF3 : END EDF SESSION
PF4 : SUPPRESS DISPLAYS     PF5 : WORKING STORAGE     PF6 : USER DISPLAY
PF7 : SCROLL BACK           PF8 : SCROLL FORWARD      PF9 : STOP CONDITIONS
PF10: PREVIOUS DISPLAY      PF11: UNDEFINED           PF12: ABEND USER TASK
```

**Finding the Program Offset**

The CEDF LINE: # is the HEX address where the macro level request was issued. This is intended to help in debugging, because OFFSET is the offset into the CICS Comet system not the application program.

To get the offset to the application program, take the LINE: # value and subtract the load point, then the CICS Comet stub length. This will give you the true offset for Cometized programs.

*LINE: # - load point = y*
*y - stub = true offset*

CICS Comet stub lengths are as follows:

- Assembler programs: X'30'
- COBOL macro-level programs: X'40'
- COBOL mixed-mode programs: X'80'
- PL/I programs: X'30'

☞ For PL/I or LE enabled programs you do not have to subtract the stub because it is placed at the end of the program. Refer to "CICS Comet Program Stub" on page 117 for a complete listing of CICS Comet stub lengths and locations.

Let's try an example. The LINE value for our sample program is 826145AA. Suppose the load point is 82610000. The program is a COBOL macro-level program, with a CICS Comet stub of X'40'. The calculation would be:

```
826145AA - 82610000 = 45AA
45AA - 40 = 456A
```

So the offset into the application program is 456A.

**Summary**

Using CEDF on macro level programs also allows you to overtype the command request or the command response shown on the screen. Examples of CEDF data that can be overwritten include the following:

- MAPSET name can be changed to correct map name.
- Map LENGTH can be changed to correct the map length.
- RESPONSE can be changed from an error condition to NORMAL response.
- Macro level data areas can be changed to new values.
- You can also delete or no-op a command level request by typing NOP over the EXEC CICS keywords. This can be very useful for disabling abend exits. They are usually set by an EXEC CICS HANDLE ABEND command, which can then be NO-OPed under CEDF.

**Finding the Abending PSW and Registers**

The contents of the PSW are displayed by CEDF at the time of the abend. Typically, the PSW points to the next instruction to be executed rather than the previous instruction that actually failed.

1. Press PF12 from the abend screen to display the abending registers.
2. Press the print key to print this screen.

   Look for the PSW following the 16 registers (0 through 15). The PSW should start at 079D*nnnn* or 078D*nnnn.* Normally*, nnnn* is usually 0000 but could be anything.

   Following this should be a fullword containing the same address as displayed on the abend screen we just came from. In the next word, the first halfword is the instruction length code (ILC) and the second halfword is the abend code. For example, 003 is the same as an MVS SOC3 abend.

3. Subtract this value from the PSW address to get the actual abending address.

   Note: It can be zero.

4. Display that address and print that screen.
5. Press PF10 to backup one page or x'100' bytes and print that screen.

   Continue to backup until you see either a copyright or a module name eye-catcher. Print as much of it as you can find. This helps us to determine the failing module name, which could be CICS Comet or the application itself.

   Following the PSW address are two halfwords and a fullword that further qualify the abend. This value can be 0000, 0002, 0004, or 0006, and is followed by a halfword ASRA exception type code.

   The exception code can be anything from 0001 to 0011, but is usually in the range of 0001 through 000F. This code can be converted to the MVS abend code by replacing the 000 with S0C.

   For example, 0001 would become S0C1, which is an operation exception. Or 0004 would be S0C4, which is a protection exception.

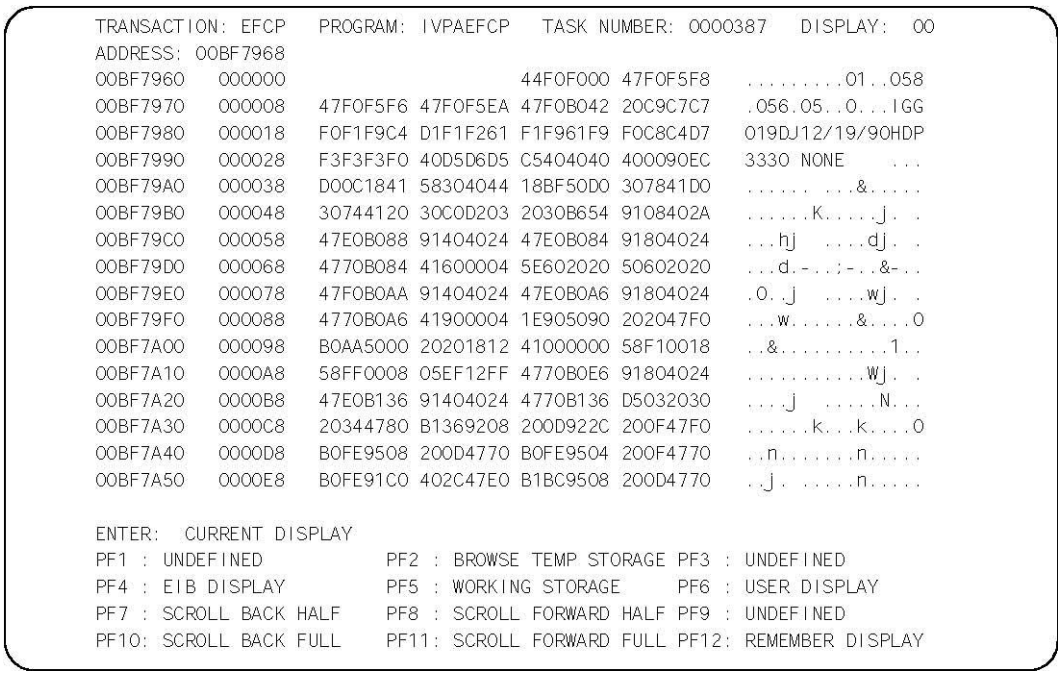
   Next comes a fullword which only has meaning if it is a protection exception, and contains the address that caused the protection exception if MVS can fetch it.

```
TRANSACTION: EFCP   PROGRAM: IVPAEFCP   TASK NUMBER: 0000387   DISPLAY:  00
STATUS:  AN ABEND HAS OCCURRED

EIBTIME      = 180736
EIBDATE      = 92275
EIBTRNID     = `EFCP'
EIBTASKN     = 387
EIBTRMID     = `L4C4'

EIBCPOSN     = 26
EIBCALEN     = 0
EIBAID       = X'7D'                                        AT X'0001C982'
EIBFN        = X'0C02'  GETMAIN                             AT X'0001C983'
EIBRCODE     = X'000000000000'                              AT X'0001C985'
EIBDS        = `........'
+   EIBREQID     = `........'
INTERRUPT: EXECUTION
ABEND :   ASRA                          PSW: X'079D0000 00BF796C 00040003'

ENTER:  CONTINUE
PF1 : UNDEFINED          PF2 : SWITCH HEX/CHAR     PF3 : END EDF SESSION
PF4 : SUPPRESS DISPLAYS  PF5 : WORKING STORAGE     PF6 : USER DISPLAY
PF7 : SCROLL BACK        PF8 : SCROLL FORWARD      PF9 : STOP CONDITIONS
PF10: PREVIOUS DISPLAY   PF11: UNDEFINED           PF12: REGISTERS AT ABEND
```

Pressing PF12 takes you to this screen with registers R0 to R15 and PSW of 078D0000 00BF7968.

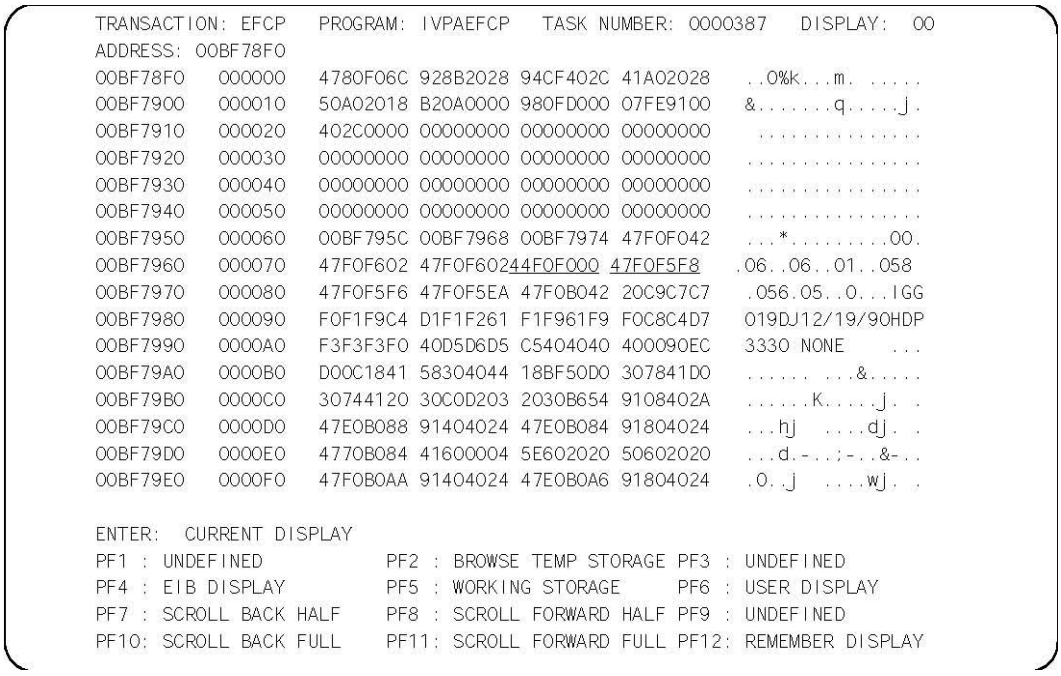
```
TRANSACTION: EFCP   PROGRAM: IVPAEFCP   TASK NUMBER: 0000387   DISPLAY:  00
ADDRESS: 0004ACC8      Reg. 0                                   Reg. 1
0004ACC0   000000                        0063F438 0063F3F8   .........4...3Q
0004ACD0   000008   0006511C 02F4AE60 00065B40 0063C260   .....4.-..$ ..B-
0004ACE0   000018   0001CA89 0001C968 00029990 00065FA4   ...i..I...r...¬u
0004ACF0   000028   0063F288 00640288 00065F10 00065010   ..2h...h..¬...&.
0004AD00   000038   8063F3A8 00BF7968 079D0000 00BF796C   ..3yk.......k...
0004AD10   000048   00040003 00000000 D7C30001 00061190   ........PC......
0004AD20   000058   00064580 02D01E20 E100F400 0000387C   .........4....@
0004AD30   000068   8064592A 798E79DC 00000000 00000000   ................
0004AD40   000078   D9C5C7E2 50D7E2E6 0063F438 0063AA40   REGS&PSW..4....
0004AD50   000088   000002B0 00049804 8C000098 0004AA40   ......q....q...
0004AD60   000098   89040058 0004ACC0 00000000 0002BF34   i..............
0004AD70   0000A8   00000000 00000000 00000000 00000000   ................
0004AD80   0000B8   405D8A82 00000000 00000000 0004A51C    ).b.........v.
0004AD90   0000C8   405D8850 0004ACC8 007C62A8 00061000    )h&...H.@.y....
0004ADA0   0000D8   00049A90 000336A4 0004A448 01048E04   .......u..u.....
0004ADB0   0000E8   01048E04 0001C800 89040058 0004ACC0   ......H.I.......

ENTER:  CURRENT DISPLAY
PF1 : UNDEFINED          PF2 : BROWSE TEMP STORAGE PF3 : UNDEFINED
PF4 : EIB DISPLAY        PF5 : WORKING STORAGE     PF6 : USER DISPLAY
PF7 : SCROLL BACK HALF   PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL   PF11: SCROLL FORWARD FULL PF12: REMEMBER DISPLAY
```

Then entering the PSW address takes us to the second screen.



```
TRANSACTION: EFCP   PROGRAM: IVPAEFCP   TASK NUMBER: 0000387   DISPLAY:  00
ADDRESS: 00BF7968
00BF7960   000000                    44F0F000 47F0F5F8   .........01..058
00BF7970   000008   47F0F5F6 47F0F5EA 47F0B042 20C9C7C7   .056.05..0...IGG
00BF7980   000018   F0F1F9C4 D1F1F261 F1F961F9 F0C8C4D7   019DJ12/19/90HDP
00BF7990   000028   F3F3F3F0 40D5D6D5 C5404040 400090EC   3330 NONE    ...
00BF79A0   000038   D00C1841 58304044 18BF50D0 307841D0   ...... ...&.....
00BF79B0   000048   30744120 30C0D203 2030B654 9108402A   ......K.....j. .
00BF79C0   000058   47E0B088 91404024 47E0B084 91804024   ...hj  ....dj. .
00BF79D0   000068   4770B084 41600004 5E602020 50602020   ...d.-..;-..&-..
00BF79E0   000078   47F0B0AA 91404024 47E0B0A6 91804024   .0..j  ....wj. .
00BF79F0   000088   4770B0A6 41900004 1E905090 202047F0   ...w......&....0
00BF7A00   000098   B0AA5000 20201812 41000000 58F10018   ..&.........1..
00BF7A10   0000A8   58FF0008 05EF12FF 4770B0E6 91804024   ...........Wj. .
00BF7A20   0000B8   47E0B136 91404024 4770B136 D5032030   ...j  .....N...
00BF7A30   0000C8   20344780 B1369208 200D922C 200F47F0   .....k...k....0
00BF7A40   0000D8   B0FE9508 200D4770 B0FE9504 200F4770   ..n.......n.....
00BF7A50   0000E8   B0FE91C0 402C47E0 B1BC9508 200D4770   ..j. .....n.....

ENTER:  CURRENT DISPLAY
PF1 : UNDEFINED          PF2 : BROWSE TEMP STORAGE PF3 : UNDEFINED
PF4 : EIB DISPLAY        PF5 : WORKING STORAGE     PF6 : USER DISPLAY
PF7 : SCROLL BACK HALF   PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL   PF11: SCROLL FORWARD FULL PF12: REMEMBER DISPLAY
```

Pressing PF7 takes you to this screen, which is a half page backwards. From here you can see the instructions and eye-catchers preceding the abending instruction.



```
TRANSACTION: EFCP   PROGRAM: IVPAEFCP   TASK NUMBER: 0000387   DISPLAY:  00
ADDRESS: 00BF78F0
00BF78F0   000000   4780F06C 928B2028 94CF402C 41A02028   ..0%k...m. .....
00BF7900   000010   50A02018 B20A0000 980FD000 07FE9100   &.......q.....j.
00BF7910   000020   402C0000 00000000 00000000 00000000     ..............
00BF7920   000030   00000000 00000000 00000000 00000000   ................
00BF7930   000040   00000000 00000000 00000000 00000000   ................
00BF7940   000050   00000000 00000000 00000000 00000000   ................
00BF7950   000060   00BF795C 00BF7968 00BF7974 47F0F042   ...*.........00.
00BF7960   000070   47F0F602 47F0F60244F0F000 47F0F5F8   .06..06..01..058
00BF7970   000080   47F0F5F6 47F0F5EA 47F0B042 20C9C7C7   .056.05..0...IGG
00BF7980   000090   F0F1F9C4 D1F1F261 F1F961F9 F0C8C4D7   019DJ12/19/90HDP
00BF7990   0000A0   F3F3F3F0 40D5D6D5 C5404040 400090EC   3330 NONE    ...
00BF79A0   0000B0   D00C1841 58304044 18BF50D0 307841D0   ...... ...&.....
00BF79B0   0000C0   30744120 30C0D203 2030B654 9108402A   ......K.....j. .
00BF79C0   0000D0   47E0B088 91404024 47E0B084 91804024   ...hj  ....dj. .
00BF79D0   0000E0   4770B084 41600004 5E602020 50602020   ...d.-..;-..&-..
00BF79E0   0000F0   47F0B0AA 91404024 47E0B0A6 91804024   .0..j  ....wj. .

ENTER:  CURRENT DISPLAY
PF1 : UNDEFINED          PF2 : BROWSE TEMP STORAGE PF3 : UNDEFINED
PF4 : EIB DISPLAY        PF5 : WORKING STORAGE     PF6 : USER DISPLAY
PF7 : SCROLL BACK HALF   PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL   PF11: SCROLL FORWARD FULL PF12: REMEMBER DISPLAY
```

# CICS Comet ASRA Abend Fact Sheet

Company name: _____

Company contact: _____

Contact phone number:_____Fax:

CICS version: _____

CICS Comet version: _____

Comet tape creation date: _____

**Information About the Abending Program**

Language: PL/I VS/COBOL ASM COBOL II BMS MAPSET

PSW: 07__D_____ _____ _____ _____

Example: 078D1000 80786046 00060004 90785042

Registers 00 - 03:_____ _____ _____ _____

Registers 04 - 07:_____ _____ _____ _____

Registers 08 - 11:_____ _____ _____ _____

Registers 12 - 15:_____ _____ _____ _____

16 bytes starting from the PSW (00786046):

(If PSW is not in the transaction dump, fax the module index from the end of the CICS transaction dump.)

32 bytes of program storage before the PSW (00786026):

Operands A and B of a failing six byte instruction and operand B of a four byte instruction:
(See “Decoding Storage Operands” on page 53, if you need help finding and decoding the machine
instruction.)

Operand A:_____ _____ _____ _____

Operand B:_____ _____ _____ _____

Look at the Module Index at the end of the dump. Find the program name in which the PSW is located, by using the load point address.

Program: _____ Load Point: _____

Length: _____ Entry Point: _____

# Decoding Storage Operands

1. Subtract the ILC from the PSW address, which is the halfword following the PSW address.

   For example, using the numbers from page 52:

   ```
   786046      The PSW from page 52
        6      The ILC from page 52

   ______
   786040      The real PSW
   ```

   This will normally be 0000, 0002, 0004, or 0006 followed by the interrupt code which will normally be 001 through 000F. The interrupt code relates directly to the system abend codes of S0C1 through S0CF.

   The ILC is the length of the instruction, unless the ILC is 0000. In this case the PSW points directly to the failing instruction. You can tell the length of the instruction by examining the first two bytes of the OPCODE. For example, if the OPCODE is X'C0' through X'FF', then the instruction is six bytes long and has two storage operands (A and B). If the OPCODE is X'40' through X'BF', the instruction is four bytes long and has only one storage operand. If the OPCODE is X'00' through X'3F', the instruction is two bytes long and has no storage operands.

   All instructions start with a one byte OPCODE followed by a one byte modifier. Four byte instructions then have a storage operand of two bytes which is formatted as follows: BDDD. B is the base register used for the storage operand, and DDD is the displacement from the address in the base register B. Six byte instructions have two storage operands, each two bytes long, formatted just like the four byte instructions above.

2. Add the displacement to the base register's address.

   Example:

   D203,C084,9180

   D2 is a MOVE CHARACTER INSTRUCTION (MVC) that is six bytes in length. 03 is the relative length of the moved data (four bytes). The first or A operand is C084, which means the base for operand A is register C or 12 with a displacement of X'084'. The second or B operand is 9180, which means the base for operand B is register 9 with a displacement of X'180'.

   If R12=00060190 and R9=00333308, then we do the following computations:

   ```
   00060190        0333308
      + 084         +   180

   ________     ________
   00060214     00342188
   ```

3. Display the storage address and print it for both operands.

   In the example described in the previous step, display addresses 0060214 and 00342188 and print the storage display screens.

# Program Debugging - Overview

**CICS Comet Domains**

Each domain converts macro requests to command level requests and then converts the command level responses to macro level responses. CC$$KERN is the mainline macro level processor, or Kernel Domain, that determines which CICS Comet domain to invoke. The other CICS Comet domains are listed below:

| CICS Function | Comet Module | Description |
|---|---|---|
| BFP | CC$$EBFP | Built-in functions |
| BMS | CC$$EBMS | Basic Mapping Services |
| DCP | CC$$EDCP | Dump control |
| DIP | CC$$EDIP | Data interchange |
| DL/I | CC$$EDLI | Data language 1 |
| FCP | CC$$EFCP | File control |
| ICP | CC$$EICP | Interval control |
| JCP | CC$$EJCP | Journal control |
| KCP | CC$$EKCP | Task control |
| PCP | CC$$EPCP | Program control |
| SCP | CC$$ESCP | Storage control |
| SPP | CC$$ESPP | Sync point program |
| TCP | CC$$ETCP | Terminal control |
| TDP | CC$$ETDP | Transient data |
| TRP | CC$$ETRP | Trace control |
| TSP | CC$$ETSP | Temporary storage |

## Exit Points within CICS Comet Domains

Each domain has two exit points, one at entry to the domain and one at exit from the domain. Exits are named CCE*x*E*nnn*, where *x* is the letter B for before entry or A for after conversion. The *nnn* suffix s the domain's name such as FCP for file control program. For example, the file control domain is CC$$EFCP and the exits are CCEBEFCP and CCEAEFCP. See "CICS Comet Exits" on page 78, for a complete list of exits.

Program exits are refreshed whenever the domain is refreshed. The exits are activated by the presence of a PPT entry for the exit. If the PPT entry is missing or disabled, CICS Comet does not activate the exit.

## How CICS Comet Gets Control

Control passes to CICS Comet when a program executes containing a CICS Comet stub. You can view the contents of the program's storage with the `HALY LOCATE` *`progname`* transaction. The CICS Comet stub can be identified by DFHYA170 in the first eight bytes and `CC$x` at the end of the stub.

The *x* character identifies the program language recognized by CICS Comet:

A Assembler,
C VS/Cobol,
P PL/I
V Assembler with a non-standard entry point. If the CC$x is missing, then the stub is a CICS command level stub. A macro level assembler program saves the true entry point offset in the full word following the CC$A and the entry address following CC$V.

If the program is mixed-mode, another stub with DFHY?nnn or DFHE follows the CICS Comet stub. VS/COBOL mixed-mode programs should also contain an RSA> following the CICS Comet stub.

## CSA/CWA

CICS Comet builds a copy of the CICS CSA and CWA at CICS Comet initialization time and reformats it to conform to the storage offsets of CICS 1.7 and 2.1.

You can view the CICS Comet CSA/CWA storage using the HALY transaction—enter HALY LOCATE COMETCSA. You can verify this is the CICS Comet CSA by pressing PF7 to see the eye-catcher CC$$CCSA.

The CICS Comet CWA follows the CSA by 512 bytes or X'200'. To see it type in an offset of `+0000200` on the HALY command line and press ENTER. Or replace COMETCSA with `COMETCWA` and press ENTER.

## TCA/TWA

CICS Comet builds a copy of the transaction TCA and TWA at CICS Comet transaction initialization time and reformats it to conform to the storage offsets of CICS 1.7 and 2.1.

To view the CICS Comet TCA/TWA storage:

1. Use the CEDF transaction and press PF5.
2. Find the CC$$EISS or CC$$EISM eye-catcher at offset x'0F8' into the EIS storage area.
3. Take the address which immediately follows at x'100' as the CICS Comet user TCA address.

   The first EIS storage area will usually be chained from the real CICS TCA at offset x'0E0' offset x'0DC' in CICS/TS.
4. The CICS Comet TWA will follow the TCA in storage by 256 bytes. So add x'100' to see the TWA. (CEDF's PF11 key usually goes forward by x'100' bytes.)

## PCT

CICS Comet builds a copy of the transaction's PCT entry at CICS Comet transaction initialization time. CICS Comet places the transaction's address in the CICS Comet system TCA and reformats it to conform to the storage offsets of CICS 1.7 and 2.1.

You can view the CICS Comet PCT by following the above instructions for finding the TCA then taking the address at offset +0 in the user TCA, which points to the CICS Comet system TCA for this transaction. The PCT address is in the same location in the CICS Comet system TCA (X'014') as it would be in a CICS 1.7 or 2.1 system.

## PPT

CICS Comet builds a copy of the transaction's PPT entry at CICS Comet transaction initialization time and places its address in the CICS Comet system TCA, and reformats it to conform to the storage offsets of CICS 1.7 and 2.1.

You can view the CICS Comet PPT by following the above instructions for finding the TCA then taking the address at offset +0 in the user TCA, which points to the CICS Comet system TCA for this transaction. The PPT address is in the same location in the CICS Comet system TCA as it would be in a CICS 1.7 or 2.1 system.

## TCTTE

CICS Comet builds a copy of the transaction's TCTTE entry at CICS Comet transaction initialization time and places its address in the CICS Comet user TCA, and reformats it to conform to the storage offsets of CICS 1.7 and 2.1.

You can view the CICS Comet TCTTE by following the above instructions for finding the TCA then taking the three byte address at offset +9 in the user TCA, which points to the CICS Comet TCTTE for this transaction.

## FCT

CICS Comet builds a copy of any FCT entry used by a CICS Comet transaction the first time the transaction makes a file request. CICS Comet reformats the FCT copy to conform to the storage offsets of CICS 1.7 and 2.1.

You can view the CICS Comet FCT by following the same procedure you would use for finding the FCT address in a true CICS FWA, VSWA, or FIOA area. These areas are in task storage, so they will be freed when the task terminates.

# THE CICS COMET STUB AND THE PROGRAM OFFSET

☞ See "Finding the Program Offset" on page 47, for information on stub length and finding the offset in a Cometized program.

# Chapter 5 Special Procedures

This chapter describes optional CICS Comet customization procedures. Most of the procedures described in this chapter are not required at most sites.

- Relinking macro-level Programs
- Installing Interfaces
- Support for COBOL II and Databases
- Coding Entries for Function Shipping
- Supporting Direct CICS Control Block Search
- ISAM Compatibility
- CICS Comet Exits

# Relinking Macro-Level Programs

Relinking is rarely necessary. However, you must relink if any of the following conditions are true:

- You do not use the Dynamic Attach facility.
  Relink your programs to append the CICS Comet program stub.


WARNING

The program must be excluded from the Dynamic Attach facility if you relink. Make sure that you remove this program from the CCMACINC Include table, or that you exclude it by adding an entry to the CCMACEXC table.

1. To relink, use the sample JCL in the CICS Comet SAMPLIB, members CBLMACRO, CBLMIXED, PLIMACRO, ASMMACRO, and ASMMIXED.

   These are JCL examples for COBOL, COBOL mixed-mode, PL/I, PL/I mixed-mode, assembler, and assembler mixed-mode. Be sure to use the correct member for the program you are relinking.
2. After you relink a COBOL program with the CICS Comet stub, you must change the PPT entry for that program to `LANGUAGE=ASSEMBLER` from COBOL.


WARNING

If you do not change the PPT entry for that program, you may experience ASRA abends in CICS module DFHPCP.

Notice the LINKEDIT `PARM='AMODE(31),RMODE(ANY)'` in the following relink JCL examples. These parameters allow the macro-level programs to run above the 16 Mb line with CICS Comet. There are only two exceptions to running above the 16 Mb line:

- Macro-level programs must run below the 16 Mb line if they depend on field TCAFCAAA or TCAFCI being set to X'01' to indicate this task is attached to a terminal.
- DL/I requires that macro-level DL/I programs provide the DL/I Segment Search Arguments (SSAs) to DL/I in non-XA storage. If your macro-level programs use qualified SSAs that are not in COBOL working storage, are hard coded in an assembler program rather than in the TWA, or are not in the PL/I DSA, then those programs must run below the line. If you attempt to run these programs above the line, DL/I ends the task with AEXT or AEXS abends. CICS Comet handles all other DL/I arguments above the 16 Mb line.

# Examples of Relink JCL

## COBOL Macro-Level: SAMPLIB(CBLMACRO)

```
//JOB@NAME JOB  1,CBLMACRO,CLASS=A,MSGCLASS=X,REGION=2048K
//*-------------------------------------------------------------
//* SAMPLE RELINK OF A COBOL MACRO LEVEL PROGRAM FOR CICS COMET
//*-------------------------------------------------------------
//LINKEDIT  EXEC    PGM=IEWL,PARM='REUS,XREF,LIST,
//                  LET,AMODE(31),RMODE(ANY)'
//SYSLIB    DD      DISP=SHR,DSN=your_old.COBOL.MACRO.LEVEL.LOADLIB
//            DD    DISP=SHR,DSN=hlq.CCV540.LOAD
//SYSUT1    DD      UNIT=VIO,SPACE=(1024,(100,50))
//SYSPRINT  DD      SYSOUT=*
//SYSLMOD DD        DISP=SHR,
//                  DSN=new.COBOL.COMMAND.LEVEL.LOADLIB(progname)
//SYSLIN    DD *
 INCLUDE SYSLIB(CC$STUBC)
 INCLUDE SYSLIB(progname)          <=== must follow include of stub
 ENTRY CBL$STUB
/*
//
```

## COBOL Mixed-Mode: SAMPLIB(CBLMIXED)

```
//JOB@NAME JOB 1,CBLMIXED,CLASS=A,MSGCLASS=X,REGION=2048K
//*-------------------------------------------------------------
//*   SAMPLE RELINK OF A MIXED-MODE COBOL PROGRAM FOR CICS COMET
//*-------------------------------------------------------------
//LINKEDIT  EXEC    PGM=IEWL,PARM='REUS,XREF,LIST,
//                  LET,AMODE(31),RMODE(ANY)'
//SYSLIB    DD      DISP=SHR,DSN=your_old.COBOL.MIXED.MODE.LOADLIB
//            DD    DISP=SHR,DSN=hlq.CCV540.LOAD
//SYSUT1    DD      UNIT=VIO,SPACE=(1024,(100,50))
//SYSPRINT  DD      SYSOUT=*
//SYSLMOD   DD      DISP=SHR,
//                  DSN=new.COBOL.COMMAND.LEVEL.LOADLIB(progname)
//SYSLIN    DD *
 INCLUDE SYSLIB(CC$STUBC)
 INCLUDE SYSLIB(CC$STUBE)
 INCLUDE SYSLIB(progname)          <=== must follow include of stub
 ENTRY CBL$STUB
/*
//
```

## PL/I Macro-Level: SAMPLIB(PLIMACRO)

```
//JOB@NAME JOB  1,PLIMACRO,CLASS=A,MSGCLASS=X,REGION=2048K
//*-------------------------------------------------------------
//* SAMPLE RELINK OF A PL/I MACRO LEVEL PROGRAM FOR CICS COMET
//*-------------------------------------------------------------
//LINKEDIT EXEC     PGM=IEWL,       PARM='REUS,XREF,LIST,LET,
//                  AMODE(31),RMODE(ANY)'
//SYSLIB     DD     DISP=SHR,DSN=your_old.PLI.MACRO.LEVEL.LOADLIB
//           DD     DISP=SHR,DSN=hlq.CCV540.LOAD
//SYSUT1     DD     UNIT=VIO,SPACE=(1024,(100,50))
//SYSPRINT DD       SYSOUT=*
//SYSLMOD DD        DISP=SHR,DSN=new.PLI.COMMAND.LOADLIB(progname)
//SYSLIN     DD *
    INCLUDE SYSLIB(progname)        <=== must precede include of stub
    INCLUDE SYSLIB(CC$STUBP)
    ENTRY PLI$STUB
/*
//
```

## PL/I Mixed-Mode: SAMPLIB(PLIMIXED)

```
//JOB@NAME JOB 1,PLIMIXED,CLASS=A,MSGCLASS=X,REGION=2048K
//*-------------------------------------------------------------
//*   SAMPLE RELINK OF A MIXED-MODE PL/I PROGRAM FOR CICS COMET
//*-------------------------------------------------------------
//LINKEDIT  EXEC    PGM=IEWL,PARM='REUS,XREF,LIST,
//                  LET,AMODE(31),RMODE(ANY)'
 //SYSLIB    DD     DISP=SHR,DSN=your_old.PLI.MIXED.MODE.LEVEL.LOADLIB
 //           DD     DISP=SHR,DSN=hlq.CCV540.LOAD
//SYSUT1    DD      UNIT=VIO,SPACE=(1024,(100,50)) //SYSPRINT DD  SYSOUT=*
//SYSLMOD DD        DISP=SHR,DSN=new.PLI.COMMAND.LOADLIB(progname)
//SYSLIN     DD *
    INCLUDE SYSLIB(progname)        <=== must precede include of stub
    INCLUDE SYSLIB(CC$STUBP)
    ENTRY PLI$STUB
/*
//
```

## Assembler Macro-Level: SAMPLIB(ASMMACRO)

```
//JOB@NAME JOB 1,ASMMACRO,CLASS=A,MSGCLASS=X,REGION=2048K
//*-------------------------------------------------------------
//*  SAMPLE RELINK OF AN ASM MACRO LEVEL PROGRAM FOR CICS COMET
//*-------------------------------------------------------------
//LINKEDIT EXEC PGM=IEWL,PARM='REUS,XREF,LIST,LET'
//SYSLIB  DD DISP=SHR,DSN=your_old.ASM.MACRO.LOADLIB
// DD DISP=SHR,DSN=hlq.CCV540.LOAD
//SYSUT1  DD UNIT=VIO,SPACE=(1024,(100,50))
//SYSPRINT DD  SYSOUT=* //SYSLMOD  DD DISP=SHR,
//          DSN=new.ASM.COMMAND.LEVEL.LOADLIB(progname)
//SYSLIN  DD *
    INCLUDE SYSLIB(CC$STUBA)
    INCLUDE SYSLIB(progname)        <=== must follow include of stub
    ENTRY ASM$STUB
/*
//
```

## Assembler Mixed-Mode: SAMPLIB(ASMMIXED)

```
//JOB@NAME JOB 1,ASMMIXED,CLASS=A,MSGCLASS=X,REGION=2048K
//*-------------------------------------------------------------
//* SAMPLE RELINK OF A MIXED-MODE ASM PROGRAM FOR CICS COMET
//*-------------------------------------------------------------
//LINKEDIT      EXEC    PGM=IEWL,PARM='REUS,XREF,LIST,LET'
//SYSLIB        DD      DISP=SHR,DSN=your_old.ASM.MIXED.MODE.LOADLIB
//              DD      DISP=SHR,DSN=hlq.CCV540.LOAD
//SYSUT1        DD      UNIT=VIO,SPACE=(1024,(100,50))
//SYSPRINT      DD      SYSOUT=* //SYSLMOD  DD DISP=SHR,
//                      DSN=new.ASM.COMMAND.LEVEL.LOADLIB(progname)
//SYSLIN  DD *
   INCLUDE SYSLIB(CC$STUBA)
   INCLUDE SYSLIB(progname)          <=== must follow include of stub
   ENTRY ASM$STUB
/*
//
```

# Installing Interfaces

**IDMS**

- IDMSARTS expects the CSA to point to the CICS DFHPCP for IDMS. IDMSARTS recognizes the Comet CSA and PCP program and does not initialize correctly. Currently, IDMSARTS is not supported. IDMS provides other alternatives.
- CICS Comet should be started after starting ACF2, but before starting IDMS.
- All application programs using the IDMS interface must be treated as mixed-mode programs and should be placed in the CCMACINC table with Y specified for the mixed-mode flag.
- This is regardless of what mode the actual application program is in (macro or command).
- The IDMS main interface module must be relinked or dynamically included in the CCMACINC table. This is normally IDMSINTC.
- CICS Comet should be placed in the DFHPLTPI prior to IDMSINTC.
- Add a PPT entry for module CC$$EINT. The PPT entry should be assembler. You can choose the RES status. On CICS/TS you may also want to turn EDF off for the PPT entry. This prevents application programmers from seeing the EXEC CICS commands being issued by CICS Comet on behalf of the database interface. The program CC$$EINT should not be included in the CCMACINC table or abends will occur, due to having two CICS Comet stubs in the program (one from UNICOM Systems' link-edit and the other from dynamic attach).
- The COBOL II working storage area must reside below the 16 Mb line to reference the database by the parmlist. DATA(24) is the COBOL II option. The Cobol II application program need not be placed in the CICS Comet macro Include table CCMACINC if you relink them with the CICS Comet supplied replacement for IDMSCINT called IDMSC$II. This is because the CICS Comet database stubs are truly command level, unlike the original database stubs which are macro-level. CC$$EINT is a macro-level program that invokes the original vendor’s database interface.
- The CICS Comet load module IDMSC$II replaces the IDMSCINT stub. You should copy IDMSC$II to the COBOL II subroutine library used during COBOL II link-edits. Then, you can rename IDMSC$II to its IDMS name, IDMSCINT, so that you don't have to change any COBOL II assembly and link JCL.

- The COBOL II programs that have the old stub in them already can be relinked with the following link edit statements:

```
//LINKEDIT   EXEC   PGM=IEWL,PARM='REUS,XREF,LIST,LET'
//SYSLIB     DD     DISP=SHR,DSN=your.old.cobol2.load.library
//          DD     DISP=SHR,DSN=hlq.CCV540.LOAD
//SYSUT1     DD     UNIT=VIO,SPACE=(1024,(100,50))
//SYSPRINT   DD     SYSOUT=*
//SYSLMOD    DD     DISP=SHR,DSN=new.cobol2.test.load.library
//SYSLIN     DD *
     REPLACE IDMSCINT
     INCLUDE SYSLIB(progname)
     INCLUDE SYSLIB(IDMSC$II)
     ENTRY progname
/*
//
```

**TOTAL/DB**

- All application programs using the TOTAL/DB interface must be specified as mixed-mode programs. A mixed-mode program must be entered in the CCMACINC table with Y specified in the mixed-mode field. This is regardless of what mode the actual application program is in (macro or command).
- The TOTAL/DB main interface module must be relinked or dynamically included in the CCMACINC table.
- CICS Comet should be placed in the DFHPLTPI prior to TOTAL/DB.
- Add a PPT entry for module CC$$ETOT. The PPT entry should be assembler, and you may choose the RES status.

  Under CICS/TS, you may also want to turn EDF off for the PPT entry. This prevents application programmers from seeing the EXEC CICS commands being issued by CICS Comet on behalf of the database interface.

- Do not include CC$$ETOT in the CCMACINC table or abends will occur. Abends occur because two CICS Comet stubs are in the programs; one from UNICOM Systems' link-edit and the other from the Dynamic Attach facility.
- TOTAL/DB COBOL II programs currently must reside below the 16 Mb line if you are using the TOTAL database interface. This is due to the interface modules issuing CICS macro-level requests. However, once the COBOL II programs are calling the CC$$ETOT interface program instead, the COBOL II programs should be able to run with RMODE=31 and DATA(24). The COBOL II working storage area must remain below for the database to reference via the parmlist.
- The TOTAL/DB COBOL II application program need not be placed in the CICS Comet macro include table CCMACINC if you relink them with the CICS Comet supplied replacement for DATBAS called DATBASII. This is because the CICS Comet database stubs are truly command level, unlike the original database stubs which are macro-level. CC$$ETOT is a macro-level program that invokes the original vendor's database interface.
- The CICS Comet load module DATBASII replaces the DATBAS stub. We recommend that you copy DATBASII to the COBOL II subroutine library used during COBOL II link-edits. Then you can rename DATBASII to its TOTAL/DB name, DATBAS, so that you don't have to change any COBOL II assembly and link JCL.
- COBOL II programs that have an existing CICS Comet stub can be relinked with the following link edit statements:

```
//LINKEDIT    EXEC   PGM=IEWL,PARM='REUS,XREF,LIST,LET'
//SYSLIB  DD  DISP=SHR,DSN=your.old.cobol2.load.library
//          DD  DISP=SHR,DSN=hlq.CCV540.LOAD
//SYSUT1      DD     UNIT=VIO,SPACE=(1024,(100,50))
//SYSPRINT    DD     SYSOUT=*
//SYSLMOD     DD     DISP=SHR,DSN=new.cobol2.test.load.library
//SYSLIN      DD *
    REPLACE DATBAS
    INCLUDE SYSLIB(progname)
    INCLUDE SYSLIB(DATBASII)
    ENTRY progname
/*
//
```

# System 2000 (S2K) Database—R11.6

- All application programs using the S2K database interface must be treated as mixed-mode and should be placed in the CCMACINC table with Y specified in the mixed-mode field. Mixed-mode is required regardless of whether the actual application program is running in macro or command mode.

- All COBOL II programs must be relinked with the S2KPLCII subroutine supplied by CICS Comet. This module can be renamed to S2KPLC and then replaced in a link-edit of the COBOL II application program. If you use S2KPLCII, then you must also add a PPT entry to the CICS Comet CSD group for CC$$ES2K as an assembler program. Failure to do so could cause APCT or AEI0 abends.

- All other application programs can be relinked with the S2KPLCII subroutine supplied by CICS Comet or dynamically attached by CICS Comet. If you use the Dynamic Attach process, you must tailor the PPT to add the CICS Comet user exit CCEACPEI and code EXITS=YES in CCSIPAM. This exit looks through the application program the first time it is loaded and replaces the S2KPLC subroutine with a updated version supplied by CICS Comet. This method makes phase-in of CICS/TS much easier than maintaining two separate load libraries, (one for 2.1 and a relinked one for CICS/TS). This exit is provide in load library format and is also provided in source format in CCV540.SAMPLIB(CCEACPEI). If the S2K code is macro level please use CCEACPEI. If some S2K code is in neither COBOL 2 nor LE, then you must use CCEACMCI.

- The S2K main interface module must be relinked with the CICS Comet command level stub or dynamically included in the CCMACINC table. This interface module is normally called CICSS2KA.

- All CICS S2K assembler modules must be reassembled and relinked using the SAS S2K.IFSOURCE library and JCL member JCLASM1, as described in the *S2K Installation Manua*l. However, you must first replace the BACKSAVE macro supplied by SAS in the IFSOURCE library with the BACKSAVE macro supplied by CICS Comet in the CCV540.MACLIB library. If you have any questions about the assembly procedure, contact the SAS S2K support group.

# Support for COBOL II and Databases

COBOL II does not support macro-level code. However, a COBOL II program can call an assembler subroutine that uses macro-level code. In this case, the COBOL II program is a mixed-mode program that requires CICS Comet support.

You can dynamically attach COBOL programs for macro-level translation. If you do not want to dynamically attach a COBOL II program, then complete the following procedure for each subroutine call.

1. Remove the COBOL II CALL statement such as:

   ```
   CALL 'macrortn' USING PARM-A, PARM-B, PARM-C.
   ```

   and replace it with the following instruction:

   ```
   CALL 'CC$$CALL' USING 'macrortn', PARM-A, PARM-B, PARM-C.
   ```

2. Recompile the COBOL II program and make sure that SYSLIB has a concatenated dataset definition for the *hlq*.CCV540.LOAD library at LINKEDIT time.

WARNING

The CC$$CALL COBOL II program and CC$$LINK must never be included in the CCMACINC MACRO level include table. If they are included in Dynamic Attach, abends may occur because CC$$LINK has already been relinked with the CICS Comet stub.

3. Add a PPT entry for the '*macrortn*' named in the COBOL II CALL statement.

   This permits the transaction using CC$$CALL and CC$$LINK to find the CALL statement at CICS execution time.

WARNING

If you do not add this PPT entry, APCT or AEI0 abends will occur.

# Coding Entries for Function Shipping

After CICS Comet converts a macro-level application program to command level, you can take advantage of CICS function shipping (also known as remote datasets). This enables programs converted by CICS Comet to use remote datasets, transient data, and temporary storage queues.

**Coding Remote DCT Entries**

An example of a remote transient data queue definition is shown below:

```
DCTCC  DFHDCT     TYPE=INITIAL,SUFFIX=CC
TDPF   DFHDCT     TYPE=REMOTE,        FIXED REMOTE                X
                  DESTID=TDPF,                                    X
                  RMTNAME=TDPF,                                   X
                  SYSIDNT=LJL1,                                   X
                  LENGTH=132          LENGTH OF FIXED LENGTH QUEUE
*
TDPV   DFHDCT     TYPE=REMOTE,        VARIABLE REMOTE             X
                  DESTID=TDPV,                                    X
                  RMTNAME=TDPV,                                   X
                  SYSIDNT=LJL1        NO LENGTH FOR VARIABLE QUEUE
       DFHDCT TYPE=FINAL
       END
```

Notice that remote entries do not indicate the queue's format. In other words, it does not specify fixed or variable. This information is critical to macro-level programs because it specifies the format of the I/O area. Variable format I/O areas expect a four byte LLBB at the beginning of the I/O area. This means if you convert a variable format intrapartition queue to a remote queue, you must not code a record LENGTH parameter on the DCT entry. If it is a fixed intrapartition queue, then you must code a LENGTH parameter for the DCT entry.

**Coding Remote FCT entries**

With CICS release 2.1, the CCFCT macro must be used in your CICS Comet FCT assembly for remote FCT entries.

For all CICS/TS releases, you must generate a CICS 2.1 FCT with the name of DFHFCTCC. Change the suffix CC to the suffix specified in CCSIPARM. CICS Comet looks for this FCT with the suffix of CC and loads it into non-XA storage. The DFHFCTCC must contain all remote entries that CICS Comet uses for function shipping file I/O requests. CICS Comet uses these entries in CICS/TS to find the RECFORM of the remote dataset (not available in CICS remote FCT entries).

CICS Comet scans the DFHFCTCC table at initialization time and completes any missing information from the real FCT table to avoid maintaining the FCT length attributes in two places.

Under CICS/TS, the FCT entries can be defined through CEDA and can be changed during a single execution of CICS. It may be necessary to refresh the CICS Comet FCT table DFHFCTCC to make sure that the two tables agree. The CICS Comet command to refresh the DFHFCTCC table is: `HALLEY REFRESH DATASETS`. This command will also NEWCOPY the DFHFCTCC

table.

☞ Use the same member for FCT definitions for function shipping, direct control block searches, and ISAM compatibility.

## Sample Remote FCT for CICS/TS Releases

```
FCTCC FCT211 TYPE=INITIAL,SUFFIX=CC
*-----------------------------------------------------------------*
*     SAMPLE REMOTE FILE DEFINITIONS FOR CICS 3.2 AND ABOVE
*-----------------------------------------------------------------*
REMOTEFB        CCFCT TYPE=REMOTE,DATASET=REMOTEFB,SYSIDNT=LJL1,      X
                RMTNAME=REMOTEFB,KEYLEN=22,LRECL=128,                 X
                RECFORM=(FIXED,BLOCKED)
REMOTEVB        CCFCT TYPE=REMOTE,DATASET=REMOTEVB,SYSIDNT=LJL1,      X
                RMTNAME=REMOTEVB,KEYLEN=22,LRECL=512,                 X
                RECFORM=(VARIABLE,BLOCKED)
REMOTEFU        CCFCT TYPE=REMOTE,DATASET=REMOTEFU,SYSIDNT=LJL1,      X
                RMTNAME=REMOTEF,KEYLEN=22,LRECL=256,                  X
                RECFORM=(FIXED,UNBLOCKED)
REMOTEVU        CCFCT TYPE=REMOTE,DATASET=REMOTEVU,SYSIDNT=LJL1,      X
                RMTNAME=REMOTEV,KEYLEN=22,LRECL=1024,                 X
                RECFORM=(VARIABLE,UNBLOCKED)    FCT211 TYPE=FINAL

 END
```

☞ Since normal CICS DFHFCT macros do not support the RECFORM parameter on remote datasets, you must use the CCFCT macro instead. When assembling your FCT, make sure the CCFCT and FCT211 macros in CICS Comet's MACLIB are accessible via the SYSLIB DD concatenation.

As with all remote dataset entries you must code an LRECL and a KEYLEN parameter on the remote FCT entry so that CICS and CICS Comet know the record length for fixed datasets and the maximum record length for variable datasets.

Examining the sample FCT above you will notice that the remote entries indicate the dataset's format, fixed or variable. This information is critical to macro-level programs as it controls the format of the I/O area. Variable format I/O areas expect a four byte LLBB at the beginning of the I/O area. If you convert a variable format VSAM dataset to a remote dataset, then you must code a RECFORM=VARIABLE parameter on the REMOTE FCT entry. If it is a fixed VSAM dataset, then you must code a RECFORM=FIXED for the REMOTE FCT entry.

You must code a BLOCKED or UNBLOCKED parameter on the remote FCT entry to control the ISAM compatibility mode. Unblocked datasets are assumed by CICS file control to be in ISAM compatibility mode.

# Supporting Direct CICS Control Block Search

**Coding Non-XA FCT entries**

If you have macro-level applications that scan CICS FCT control blocks without using the standard DFHFC CTYPE macro SPI and are running CICS Comet on a CICS/TS system, then you must define those files in a separate CICS Comet FCT.

1. Scan your source programs for the occurrence of CSAFCTBA or FCTDSTEL to determine if your programs make a direct access to FCT entries.

   Use the same member for FCT definitions for function shipping, direct control block searches, and ISAM compatibility.

   Add the file definitions to CICS Comet SAMPLIB member DFHFCTCC.
   The FCT table defaults to a suffix of NO (see SAMPLIB member CCSIPARM). Explicitly make this suffix CC (or any other two characters except NO or YE.

   The suffix should not match any real table name/suffix.

   Assemble and link edit member DFHFCTCC into the CICS Comet load library to activate the non-XA FCT entries.

   ☞ When assembling your FCT make sure the FCT211 macro in the CICS Comet MACLIB is accessible via the SYSLIB DD concatenation.

2. Create a PPT entry for DFHFCT&suffix or an RDO entry, if you are not using tables.

CICS Comet loads this FCT table into non-XA storage at CICS Comet initialization time and completes any missing information from the real FCT table. so that you do not have to maintain the FCT length attributes in two places.

```
FCTCC    FCT211 TYPE=INITIAL,SUFFIX=CC
*-----------------------------------------------------------------*
*    SAMPLE NON-XA FILE DEFINITIONS
*-----------------------------------------------------------------*
IVPFILEA  FCT211        TYPE=DATASET,DATASET=IVPFILEA,              X
                        ACCMETH=VSAM,RECFORM=(FIXED,BLOCKED),       X
                        DISP=SHR,DSNAME=HLQ.COMET.CCV540.IVPFILEA,X
                        SERVREQ=(UPDATE,ADD,BROWSE,DELETE),         X
                        BUFNI=3,BUFND=7,STRNO=1,LSRPOOL=NONE,       X
                        FILSTAT=({CLOSED|OPEN},ENABLED)
          FCT211 TYPE=FINAL
END
```

Under CICS/TS, the FCT entries can be defined through CEDA and can therefore be changed during a single execution of CICS. So it may become necessary to refresh the CICS Comet FCT table DFHFCTCC so that the two tables agree. The CICS Comet command to refresh the DFHFCTCC table is:

```
HALLEY REFRESH DATASETS
```

This command will also NEWCOPY the DFHFCTCC table.

**Coding Non-XA DCT entries**

You must code separate transient data definitions in a CICS Comet DCT under the following conditions:

- You have macro-level applications that scan through CICS DCT control blocks without using the standard DFHTD CTYPE macro SPI
- You are running CICS Comet on a CICS/TS system.

1. Scan your source programs for the occurrence of CSADCTBA or TDDCTELN to determine if your programs make a direct access to DCT entries.
2. Add the transient data definitions to the CICS Comet DFHDCTCC SAMPLIB member.
3. Change the DCT table suffix to CC or any other two characters except NO or YE.

   The DCT table defaults to a NO suffix. See the CCSIPARM SAMPLIB member. The suffix should not match any real table name/suffix.

4. Assemble and link edit the DFHDCTCC member into the CICS Comet load library to activate non-XA DCT entries.

   Make sure the CICS 2.1 DFHDCT macro is accessible via the SYSLIB DD concatenation when assembling your DCT.

5. Create a PPT entry for DFHDCTCC or an RDO entry, if you are not using tables.

```
DCTCC DFHDCT TYPE=INITIAL,SUFFIX=CC
*----------------------------------------------------------------*
*    SAMPLE NON-XA TRANSIENT DATA DEFINITIONS
*----------------------------------------------------------------*
TDPF  DFHDCT TYPE=REMOTE,  FIXED REMOTE        X
      DESTID=TDPF,                             X
      RMTNAME=TDPF,                            X
      SYSIDNT=LJL1,                            X
      LENGTH=132  LENGTH OF FIXED LENGTH QUEUE
*
TDPV  DFHDCT TYPE=REMOTE,  VARIABLE REMOTE     X
      DESTID=TDPV,                             X
      RMTNAME=TDPV,                            X
      SYSIDNT=LJL1  NO LENGTH FOR VARIABLE LENGTH QUE
 DFHDCT TYPE=FINAL
 END
```

If you modify the real DCT table or add another entry to the CICS Comet DCT table, you must refresh the CICS Comet DCT table DFHDCTCC so that the two tables agree.

The CICS Comet command to refresh the DFHDCTCC table is:

```
HALLEY REFRESH TRANSDATA
```

This command will also NEWCOPY the DFHDCTCC table.

**Coding Non-XA JCT entries**

If you have macro-level applications that scan through CICS JCT control blocks by using the DFHJC CTYPE SPI requests and are running CICS Comet on a CICS/TS system, then you must create entries for those journals in a separate CICS Comet JCT.

1. Scan your source programs for the occurrence of CSAJCTBA to determine if your programs directly access the JCT entries.
2. Add the journal definitions to the DFHJCTCC member of the SAMPLIB dataset.
3. Change the JCT table suffix to `CC` or any other two characters except NO or YE.

   The suffix should not match any real table name/suffix.

   The JCT table defaults to a suffix of NO (see SAMPLIB member CCSIPARM).
4. Assemble and link edit the DFHJCTCC member into the CICS Comet load library to activate non-XA JCT entries.

   ☞ When assembling your JCT make sure that a CICS 2.1 DFHJCT macro is accessible via the SYSLIB DD concatenation.

5. Create a PPT entry for DFHJCTCC or an RDO entry if you are not using tables.

```
JCTCC DFHJCT TYPE=INITIAL,SUFFIX=CC
*-----------------------------------------------------------------*
*    SAMPLE NON-XA JOURNAL DATASET DEFINITIONS
*-----------------------------------------------------------------*
DFHJCT TYPE=ENTRY,JFILEID=SYSTEM,  CICS SYSTEM LOG          X
      BUFSIZE=19069,                                        X
      BUFSUV=8192,                                          X
      JOUROPT=(CRUCIAL,INPUT,RETRY),                        X
      JTYPE=DISK2
DFHJCT TYPE=ENTRY,JFILEID=2,    CICS SMF LOG X
      BUFSIZE=8248,                                         X
      BUFSUV=4000,                                          X
      FORMAT=SMF,                                           X
JTYPE=SMF
DFHJCT TYPE=FINAL
END
```

If you modify the real JCT table, or add another entry to the CICS Comet JCT table, you must refresh the CICS Comet JCT table DFHJCTCC so that the two tables agree. The CICS Comet command to refresh the DFHJCTCC table is:

```
HALLEY REFRESH JOURNALS
```

This command will also NEWCOPY the DFHJCTCC table.

**Coding Non-XA PCT entries**

If you have macro-level applications that scan through CICS PCT control blocks without using the standard DFHKC CTYPE macro SPI and are running CICS Comet on a CICS/TS system, then you must create entries for those transactions in a separate CICS Comet PCT.

1. Scan your source programs for the occurrence of CSAPCTBA or PCTELN to determine if your programs directly access PCT entries.
2. Add the transaction definitions to CICS Comet SAMPLIB member DFHPCTCC.
3. Change the suffix of the PCT table to `CC` or any other two characters except NO or YE. The suffix should not match any real table name/suffix.

   The PCT table defaults to a suffix of NO (see SAMPLIB member CCSIPARM).
4. Assemble and link edit the DFHPCTCC member into the CICS Comet load library to activate non-XA PCT entries.

   ☞ When assembling your PCT make sure that a CICS 2.1 DFHPCT macro is accessible via the SYSLIB DD concatenation.

5. Create a PPT entry for DFHPCTCC or an RDO entry, if you are not using tables.

CICS Comet loads the DFHPCTCC table at CICS Comet initialization time and completes any missing information from the real PCT table.

```
    DFHPCT TYPE=INITIAL,SUFFIX=CC
*--------------------------------------------------------------*
*   SAMPLE NON-XA TRANSACTION DEFINITIONS
*--------------------------------------------------------------*
    DFHPCT TYPE=INITIAL,SUFFIX=CC
    DFHPCT TYPE=ENTRY,TRANSID=HALE,PROGRAM=CC$$STRT,TWASIZE=0
    DFHPCT TYPE=ENTRY,TRANSID=KC$$,PROGRAM=CCKC$$AT,TWASIZE=12288
    DFHPCT TYPE=ENTRY,TRANSID=KCCC,PROGRAM=CCKCE$AT,TWASIZE=12288
    DFHPCT TYPE=FINAL
    END DFHPCTBA
```

If you modify the real PCT table or add another entry to the CICS Comet PCT table, you must refresh the CICS Comet PCT table DFHPCTCC so that the two tables agree. The CICS Comet command to refresh the DFHPCTCC table is:

```
HALLEY REFRESH TRANSACTIONS
```

This command will also NEWCOPY the DFHPCTCC table.

**Coding Non-XA PPT entries**

If you have macro-level applications that scan through CICS PPT control blocks without using the standard DFHPCC CTYPE macro SPI and are running CICS Comet on a CICS/TS system, then you must create entries for those programs in a separate CICS Comet PPT.

1. Scan your source programs for the occurrence of CSAPPTBA or PPTELN to determine if your programs directly access PPT entries.
2. Add the program definitions to the DFHPPTCC member of the SAMPLIB dataset.
3. Change the suffix of the PPT table to CC or any other two characters except NO or YE.

   The suffix should not match any real table name/suffix.

   The PPT table defaults to a suffix of NO (see SAMPLIB member CCSIPARM).
4. Assemble and link edit the DFHPPTCC member into the CICS Comet load library to activate non-XA PPT entries.

☞ When assembling your PPT make sure that a CICS 2.1 DFHPPT macro is accessible via the SYSLIB DD concatenation.

5. Create a PPT entry for DFHPPTCC or an RDO entry, if you are not using tables.

CICS Comet loads the DFHPPTCC table at CICS Comet initialization time and completes any missing information from the real PPT table.

```
    DFHPPT  TYPE=INITIAL,SUFFIX=CC
*---------------------------------------------------------------*
*    SAMPLE NON-XA PROGRAM DEFINITIONS
*---------------------------------------------------------------*
    DFHPPT  TYPE=ENTRY,PROGRAM=CCKC$$AT
    DFHPPT  TYPE=ENTRY,PROGRAM=CCKCE$AT
    DFHPPT  TYPE=FINAL
   END DFHPPTBA
```

If you modify the real PPT table or add another entry to the CICS Comet PPT table, you must refresh the CICS Comet PPT table DFHPPTCC so that the two tables agree. The CICS Comet command to refresh the DFHPPTCC table is:

```
HALLEY REFRESH PROGRAMS
```

This command will also NEWCOPY the DFHPPTCC table.

**Coding Non-XA TCT entries**

If you have macro-level applications that scan through CICS TCT control blocks without using the standard DFHTC CTYPE macro SPI and are running CICS Comet on a CICS/TS system, then you must create entries for those terminals in a separate CICS Comet TCT.

1. Scan your source programs for the occurrence of CSATCTBA or TCTTETEL to determine if your programs directly access TCT entries.
2. Add the terminal definitions to CICS Comet SAMPLIB member DFHTCTCC.
3. Change the suffix of the TCT table to CC or any other two characters except NO or YE.

   The suffix should not match any real table name/suffix. The TCT table defaults to a suffix of NO. Refer to the CCSIPARM member of the SAMPLIB dataset.
4. Assemble and link edit the DFHTCTCC member into the CICS Comet load library to activate non-XA TCT entries.

☞ When assembling your TCT make sure the Comet MACLIB is accessible via the SYSLIB DD concatenation.

5. Create a PPT entry for DFHTCTCC or an RDO entry if you are not using tables.

```
TCTCC DFHTCT TYPE=INITIAL,SUFFIX=CC,ACCMETH=VTAM
*----------------------------------------------------------------*
*    SAMPLE NON-XA TERMINAL DEFINITIONS
*----------------------------------------------------------------*
L002  DFHTCT       TYPE=TERMINAL,  3277 MOD 2                  X
                   TRMIDNT=L002,                               X
                   NETNAME=NODEL002,                           X
                   ACCMETH=VTAM,                               X
                   PGESTAT=PAGE,                               X
                   FEATURE=(DCKYBD,UCTRAN,3270E,SELCTPEN),     X
                   GMMSG=YES,                                  X
                   RELREQ=(YES,YES),                           X
                   TIOAL=(2048,4096),                          X
                   TRMPRTY=10,                                 X
                   TRMMODL=2,                                  X
                   TRMSTAT=TRANSCEIVE,                         X
                   TRMTYPE=L3277
         DFHTCT TYPE=FINAL
         END
```

If you modify the real TCT table or add another entry to the CICS Comet TCT table, you must refresh the CICS Comet TCT table DFHTCTCC so that the two tables agree. The CICS Comet command to refresh the DFHTCTCC table is:

```
HALLEY REFRESH TERMINALS
```

This command will also NEWCOPY the DFHTCTCC table.

**Coding Non-XA TST entries**

If you have macro-level applications that scan through CICS TST control blocks by using the DFHTS CTYPE macro SPI and are running CICS Comet on a CICS/TS system, then you must create entries for those temporary storage queues in a separate CICS Comet TST.

1. Scan your source programs for the occurrence of CSATSTBA to determine if your programs directly access TST entries.
2. Add the queue definitions to CICS Comet SAMPLIB member DFHTSTCC.
3. Change the suffix of the TST table to CC or any other two characters except NO or YE.

   The suffix should not match any real table name/suffix.

   The TST table defaults to a suffix of NO (see SAMPLIB member CCSIPARM).
4. Assemble and link edit the DFHTSTCC member into the CICS Comet load library to activate non-XA TST entries.

☞ When assembling your TST make sure that a CICS 2.1 DFHTST macro is accessible via the SYSLIB DD concatenation.

5. Create a PPT entry for DFHTSTCC or an RDO entry, if you are not using tables.

CICS Comet will load the DFHTSTCC table at CICS Comet initialization time and fill in missing information from the real TST table, so that you do not have to maintain the TST attributes in two places.

```
TSTCC DFHTST  TYPE=INITIAL,SUFFIX=CC
*---------------------------------------------------------------*
*    SAMPLE NON-XA TEMPORARY STORAGE TABLE DEFINITIONS
*---------------------------------------------------------------*
         DFHTST TYPE=REMOTE,DATAID=RMTE,SYSID=LJL3
         DFHTST TYPE=RECOVERY,DATAID=REC
         DFHTST TYPE=FINAL
         END
```

If you modify the real TST table or add another entry to the CICS Comet TST table, you must refresh the CICS Comet TST table DFHTSTCC so that the two tables agree. The CICS Comet command to refresh the DFHTSTCC table is:

```
HALLEY REFRESH TEMPSTOR
```

This command will also NEWCOPY the DFHTSTCC table.

# ISAM Compatibility

CICS/TS no longer supports unblocked VSAM as ISAM in compatibility mode. You must create a CICS Comet pseudo FCT and specify the file as unblocked if your are using ISAM compatibility in a non-CICS/TS FCT.

1. Add the file definitions to CICS Comet SAMPLIB member DFHFCTCC.

   Use the same member for FCT definitions for function shipping, direct control block searches, and ISAM compatibility.

2. The FCT table defaults to a suffix of NO (see SAMPLIB member CCSIPARM). Explicitly make this suffix CC or any other two characters except NO or YE.

   The suffix should not match any real table name/suffix.

3. Assemble and link edit the DFHFCTCC member into the CICS Comet load library to activate non-XA FCT entries.

   Make sure the FCT211 macro in the CICS Comet MACLIB is accessible via the SYSLIB DD concatenation when assembling your FCT.

4. Create a PPT entry for DFHFCT&suffix or an RDO entry, if you are not using tables.

   CICS Comet loads this FCT table into non-XA storage at CICS Comet initialization time and fill in missing information from the real FCT table, so that you do not have to maintain the FCT length attributes in two places.

```
FCTCC    FCT211 TYPE=INITIAL,SUFFIX=CC
*--------------------------------------------------------------*
*    SAMPLE NON-XA FILE DEFINITIONS
*--------------------------------------------------------------*
IVPFILEA     FCT211          TYPE=DATASET,DATASET=IVPFILEA,          X
                     ACCMETH=VSAM,RECFORM=(FIXED,UNBLOCKED),         X
                     DISP=SHR,DSNAME=hlq.COMET.CCV540.IVPFILEA,      X
                     SERVREQ=(UPDATE,ADD,BROWSE,DELETE),             X
                     BUFNI=3,BUFND=7,STRNO=1,LSRPOOL=NONE,           X
                     FILSTAT=({CLOSED|OPEN},ENABLED)
                FCT211 TYPE=FINAL

   END
```

In CICS/TS, the FCT entries may be defined through CEDA and can therefore be changed during a single execution of CICS. So it may become necessary to refresh the CICS Comet FCT table DFHFCTCC so that the two tables agree. The CICS Comet command to refresh the DFHFCTCC table is:

```
HALLEY REFRESH DATASETS
```

This command will also NEWCOPY the DFHFCTCC table.

# CICS Comet Exits

The following table briefly describes CICS Comet exits. The table includes exit names, the CICS Comet domain they are related to, and a description of when they are called. Exits have been created in pairs; one called before domain processing, the other after domain processing. All CICS Comet exit names begin with CCE. Since version 3.3.1 Comet no longer functions with CC$$STUB. If you are executing this exit CCE?STUB exit it will not be recognized by COMET. However the CCE?STUB exit may likely function as before by renaming CCE?STUB to CC??CMCI. All USER exits are to be maintained by the Customer and updated to be compatible with changes in CICS.

Sample exits for this release can be found as members of the CICS Comet SAMPLIB dataset. Exit members are identified with a CCE prefix.

| Exit Type | Exit Name | Description and CICS Comet Calling Module |
|---|---|---|
| Application Initialization | CCEBCPEI | Before CICS Comet determines language specifications |
| | CCEACPEI | After CICS Comet has built a copy of the TCA/TWA |
| Macro-level statement intercept (global exit points) | CCEBKERN | Before CICS Comet converts the macro request to command level. |
| | CCEAKERN | After CICS Comet receives the response from command level. |
| | These two exits are invoked for every macro-level request handled by CICS Comet. Therefore, care should be taken not to add too many instructions at this point. It is more efficient to use a specific exit such as CCE?EFCP to intercept all file control macro requests than to use this global exit point. | |
| Basic Mapping | CCEBEBMS | Before CICS Comet CC$$EBMS is started |
| | CCEAEBMS | After CICS Comet CC$$EBMS is completed |
| BMS Fast Function Mapping (2.1 only) | CCEBBMSX | Before CICS Comet moves a BMS Mapset to XA storage |
| | CCEABMSX | After CICS Comet has determined to move a BMS Mapset to XA storage |
| BMS Full Function Mapping (2.1 only) | CCEBBMSP | Before CICS Comet moves a BMS Mapset to XA storage |
| | CCEABMSP | After CICS Comet has determined to move a BMS Mapset to XA storage |
| Built In Functions | CCEBEBFP | Before CICS Comet CC$$EBFP is started |
| | CCEAEBFP | After CICS Comet CC$$EBFP is completed |
| CC$$CMCI | CCEBECBI | CMCI is the CICS control block initializer, and is called from macro level init “CC$$CPEI” or from COBOL 2 init |

| | | |
|---|---|---|
| | | "CC$$C2LI" or from the LE initialization "CC$$CCEE". Before Comet macro-level control block initialization only. LE is required. |
| | CCEAECBI | After CICS Comet macro-level control block initialization only if LE is active. |

| Exit Type | Exit Name | Description and CICS Comet Calling Module |
|---|---|---|
| Data Interchange | CCEBEDIP | Before CICS Comet CC$$EDIP is started |
| | CCEAEDIP | After CICS Comet CC$$EDIP is completed |
| DL/I | CCEBEDLI | Before CICS Comet CC$$EDLI is started |
| | CCEAEDLI | After CICS Comet CC$$EDLI is completed |
| Dump Control | CCEBEDCP | Before CICS Comet CC$$EDCP is started |
| | CCEAEDCP | After CICS Comet CC$$EDCP is completed |
| EXEC CICS ADDRESS CSA | CCEBEEIP | Before CICS Comet scans the Include/Exclude Tables |
| | CCEAEEIP | After CICS Comet has scanned the Include/Exclude Tables |
| EXEC FILTER | CCEBEXEC | Before execution of mixed-mode command level statement |
| | CCEAEXEC | After execution of mixed-mode command level statement |
| File Control | CCEBEFCP | Before CICS Comet CC$$EFCP is started |
| | CCEAEFCP | After CICS Comet CC$$EFCP is completed |
| Interval Control | CCEBEICP | Before CICS Comet CC$$EICP is started |
| | CCEAEICP | After CICS Comet CC$$EICP is completed |
| Journal Control | CCEBEJCP | Before CICS Comet CC$$EJCP is started |
| | CCEAEJCP | After CICS Comet CC$$EJCP is completed |
| Kernel | CCEBDATE | CCEBDATE can be used to control the CSA date format in the CICS Comet CSA. This exist is called just before CICS Comet refreshes the CSA date and the time fields in its CSA from the real CSA. Return is by Register 14. At entry, Register 1 points to a PARMLIST that contains the TCA address at offset 0 and the CSA address at offset 4. The actual date and time fields are at offset 12. |

| | | |
|---|---|---|
| Program Control | CCEBEPCP | Before CICS Comet CC$$EPCP is started |
| | CCEAEPCP | After CICS Comet CC$$EPCP is completed |

| Exit Type | Exit Name | Description and CICS Comet Calling Module |
|---|---|---|
| Storage Control | CCEBESCP | Before CICS Comet CC$$ESCP is started |
| | CCEAESCP | After CICS Comet CC$$ESCP is completed |
| Sync Point | CCEBESPP | Before CICS Comet CC$$ESPP is started |
| | CCEAESPP | After CICS Comet CC$$ESPP is completed |
| Task Control | CCEBEKCP | Before CICS Comet CC$$EKCP is started |
| | CCEAEKCP | After CICS Comet CC$$EKCP is completed |
| Temporary Storage | CCEBETSP | Before CICS Comet CC$$ETSP is started |
| | CCEAETSP | After CICS Comet CC$$ETSP is completed |
| Terminal Control | CCEBETCP | Before CICS Comet CC$$ETCP is started |
| | CCEAETCP | After CICS Comet CC$$ETCP is completed |
| Trace Program | CCEBETRP | Before CICS Comet CC$$ETRP is started |
| | CCEAETRP | After CICS Comet CC$$ETRP is completed |
| Transient Data | CCEBETDP | Before CICS Comet CC$$ETDP is started |
| | CCEAETDP | After CICS Comet CC$$ETDP is completed |

**How to Use the Exits**

Use extreme caution testing your code if you must use a CICS Comet exit. Make sure the exit operates under all conditions that occur in your data center.

Each sample exit includes a simple return branch. If you create an exit program and want to use it with an already operational CICS Comet region: there is a sample procedure to assemble the exits in the ASMEXITS member located in the SAMPLIB dataset.

1. Add a PPT entry for it and NEWCOPY it from your CICS load library.
2. Load it with option HOLD. You can use CECI LOAD PROGRAM('progname') HOLD to do this.
3. Enter `HALLEY REFRESH` *program* from any CICS terminal to reload the CICS Comet program with your exits.

   The program name must start with CC$$. The only exception is CCEBDATE, which can be directly refreshed.

# Appendix A. About CICS Comet

## Restrictions and Statement of Support

- CICS Comet supports the following SPI macros:
  DFHTM (See note below.)
  DFHOC (Documented in the *CICS Customization Guide*.)
  DFHTD CTYPE=LOCATE or CTYPE=BROWSE (DCT)
  DFHJC CTYPE=LOCATE or CTYPE=BROWSE (JCT)
  DFHFC CTYPE=LOCATE or CTYPE=BROWSE (FCT)
  DFHKC CTYPE=LOCATE or CTYPE=BROWSE (PCT)
  DFHPC CTYPE=LOCATE or CTYPE=BROWSE (PPT)
  DFHTC CTYPE=LOCATE or CTYPE=BROWSE (TCT)
  DFHTS CTYPE=LOCATE or CTYPE=BROWSE (TST)

- CICS Comet does not support programs that are CICS management module exit points, even if they issue macro level requests. These programs must be rewritten. If you need assistance, UNICOM Systems will rewrite them on contract.
  CICS exits such as XKCREQ or XSCREQ are enabled by a command level program:

  EXEC CICS ENABLE EXIT('XSCREQ') PROGRAM('*progname*') START

  This statement of support applies to both G.L.U.E. and T.R.U.E. exit points.

- If you relink a COBOL program with the CICS Comet stub, you must also change the PPT entry to LANGUAGE=ASSEMBLER from COBOL. Dynamic Attach does this for you automatically.

  Note: This protects your investment in VS/COBOL programs when CICS will no longer support VS/COBOL.


WARNING

If the language parameter is not changed for VS/COBOL, you may experience ASRA abends in CICS DFHPCP.

- When macro level issues a TYPE=GET for DFHTD TYPE=REMOTE queues, the TDIA area coming back will look like an extrapartition queue. The address in the TCATDAA points to an LLBB for variable length records and directly to the first byte of data for fixed length records while intrapartition expects a 36/40 byte prefix. There is a CICS Comet exit, CCEAETDP, that can be used to modify this prefix length. See CCEAETDP in SAMPLIB for an example.
- For DFHDC TYPE=SEGMENT with TCASCNB=X‘0000’, CICS Comet will produce one extra transaction dump for each segment area in the segment list.

- DFHTM macro requests are supported only if the 16 byte parmlist is followed by another

20 bytes for saving registers 12 through 0. The parmlist must be increased from 16 to 36 bytes. Note that CICS 4.1 removed support for PCT and PPT entries from DFHTM.

- (CICS41 and above 20-byte parm list). These programs should be changed to use the CTYPE=LOCATE request parameters with DFHPC and other macros.

- DATALOC=ANY may be specified on the PCT and PPT entries for Cometized macro application programs only if the RMODE flag in CCMACINC was set to yes (Y), or if the program was relinked with the CICS Comet command level stub and RMODE=ANY, AMODE=31 was specified in the link edit PARM field.

☞ Macro level applications dependent on TCAFCI or that issue DFHFC TYPE=SETL in locate mode instead of move mode must run below the 16M line.

- CICS Comet has the same restriction as COBOL II regarding 31-bit execution. If you link an assembler subroutine that invokes 24-bit SVCs into a COBOL program, the program must be linked below the 16 Mb line.

# Messages and Codes

CCLL00?W: **applid*: CICS COMET REMOVED PROGRA M: *progname* FROM RESIDENCY

Explanation: The *? i*s the language type of the application program that was resident before CICS Comet added the command level stub to the beginning of the program. The *progname* is the program name in the PPT. See “Relinking Macro-Level Programs” on page 58, for more information.

CCLL000I: **applid*: CICS COMET DYNAMICALLY ATTACHED ????? PROGRAM: *xxxxxxxx*

Explanation: Normal informational message for Dynamic Attach. ????? will be COBOL, PL/1, or ASM.

CCLL000W: **applid*: CICS COMET BYPASSED ATTACHING PROGRAM: *xxxxxxxx*

Explanation: Warning message. Program name is in Include table for Dynamic Attach, but CICS Comet was not able to attach to this program. Program may be command level or mixed-mode and does not have “force” set to Y (yes).

CCLL001I: **applid*: CICS COMET TERMINATION STAGE *n*

Explanation: CICS and CICS Comet are shutting down. The message indicates the current CICS termination stage *n* at the time the message is issued.

CCLL001I: **applid*: CICS COMET INITIALIZATION COMPLETE

Explanation: Normal start-up informational message.

CCLL002I: **applid*: CICS COMET 31-BIT BMS MAPSET INITIALIZATION COMPLETE

Explanation: Normal start-up informational message for CICS /TS.

CCLL004I: CICS COMET ??????? INITIALIZATION COMPLETE Explanation Normal start-up informational message. ?????? is the initialized language: COBOL/VS VS/COBOL I LE COBOL/LE CBL2 COBOL II PL/I PL/I

CCLL005W: **applid*: CICS COMET LOADED WRONG VERSION OF MODULE CC$$CPEI

CCLL005W: **applid*: CC$$STRT VERSION: V?R?M? CC$$???? VERSION: V?R?M?

Explanation: Warning: CICS Comet start-up has discovered incompatible releases of CICS Comet modules in the execution load libraries. Reinstall CICS Comet and delete modules from the previous release. CICS

Comet will attempt to continue execution.

CCLL006I: **applid**: CICS COMET EXIT: CCE????? IS NOW ACTIVATED

Explanation: Normal start-up informational message if CICS Comet exits are installed.

CCLL007E: **applid**: CICS COMET - THE VS/COBOL LIBRARY IS MISSING FROM STEPLIB

Explanation: The VS/COBOL library must be in the STEPLIB or link list to run VS/COBOL programs.

CCLL011E: **applid**: CICS COMET INITIALIZATION ERROR

Explanation: Check that all PPT and PCT entries are installed in the PPT and PCT tables. Make sure that all INCLUDE/EXCLUDE DDs and the start-up parameter, PRVMOD=(DFHEIP), are present in the CICS start-up JCL.

CCLL012E: **applid**: CICS COMET BMS CC$$BMS? INITIALIZATION ERROR

Explanation: Verify the PPT entry for CC$$BMS? is in the PPT table.

CCLL013I: **applid**: CICS COMET BMS NOT USABLE FOR THIS CICS/TS RELEASE

Explanation: CICS Comet support for BMS maps is not usable on non-TS releases of CICS.

CCLL014E: **applid**: CICS COMET DETECTED A MISMATCH IN THE AICB ADDRESS

Explanation: This message indicates that another vendor application has altered the CICS control block called the DFHAICB. To avoid this message, you must run the CICS/TS pre-initialization processor, CC$$V540. See “Dynamic Attach for Macro-Level Translation” on page 17, for instructions.

CCLL015E: **applid**: CICS COMET LOAD MODULE CC$$???? LOAD ERROR

Explanation: Check that the PPT entry for CC$$???? is in PPT table, program is not disabled, or that the load library has been compressed.

CCLL015S: **applid**: CICS COMET CC$$CPEI INTERNAL INITIALIZATION ERROR

Explanation: The CWA length or CWA address is missing or invalid. Or GETMAIN for the CICS Comet CSA/CWA failed. Contact UNICOM Systems.

CCLL017W: *applid*: ERROR ENCOUNTERED REFRESHING PROGRAM: ????????

Explanation: A processing program table entry in the load library member DFHPPTCC does not exist in the real CICS processing program table (PPT). Check your assembly of DFHPPTCC and reassemble if necessary. After the reassembly, refresh the programs by issuing the CICS Comet command: `HALLEY REFRESH PROGRAMS`.

CCLL018W: *applid*: ERROR ENCOUNTERED REFRESHING DATASET: ????????

Explanation: A file control table entry in the load library member DFHFCTCC does not exist in the real CICS file control table (FCT). Check your assembly of DFHFCTCC and reassemble if necessary. After the reassembly, refresh the datasets by issuing the `HALLEY REFRESH DATASETS` command.

CCLL019E: *applid*: CICS COMET ALREADY INITIALIZED

Explanation: This message indicates that you have started CICS Comet previously. This message could also follow message CCLL011E after fixing the cause of that error.

CCLL024E: *applid*: CICS COMET DYNAMIC ATTACH SELF-TEST FAILED, FEATURE DISABLED

Explanation: Dynamic Attach failed during a test program load. Dynamic Attach is disabled and CICS Comet processing is disabled for all programs that are dynamically attached. To correct this problem, check Dynamic Attach installation. Check to see if maintenance has been done to this CICS region, and check the block size for DFHRPL—it should be as large as the largest block size in the concatenation.

CCLL030E: *applid*: CICS COMET COULD NOT FIND THE CICS LOADER

Explanation: CICS Comet cannot find the CICS Loader to dynamically attach programs. Check your CICS start-up JCL.

CCLL031E: *applid*: CICS COMET MVS/ESA PRE-INITIALIZATION WAS NEVER PERFORMED

Explanation: CICS Comet (CC$$V540) has not been started by MVS/ESA initialization. Review the section on dynamically attaching macro level programs.

CCLL031E: *applid*: CICS COMET COULD NOT FIND THE CICS/TS LOADER

Explanation: Check for other products that modify the MVS TCBs or CICS AFCBs.

CCLL032E: *applid*: CICS COMET COULD NOT FIND TABLE: *tablename*

Explanation The *tablename* table is not in the PPT or in RPL load library concatenation. Correct and restart CICS Comet.

CCLL033E: *applid*: CICS COMET FOUND TABLE: *module* TO BE RELEASE: *????*

Explanation The table *module* was assembled with the wrong CICS release. (You must use CICS Version 1.7 or 2.1.)

CCLL034E: *applid*: CICS COMET FOUND TABLE: *module* TO BE: *????????*

Explanation The table *module* is missing, or it contains the wrong module prefix. *????????* is the module prefix CICS Comet found.

CCLL035E: *applid*: CICS COMET FOUND TABLE: *module* TO BE LINKED INCORRECTLY

Explanation The entry point to the table should be specified on the assembler END statement. For example: END DFHFCTBA.

CCLL082I: *product ,* PRODUCT REFRESH SUCCESSFUL Explanation CC$$V540 has successfully run and the *product* was refreshed.

CCLL100I: *applid*: CICS COMET DYNAMICALLY ATTACHED PROGRAM: *progname*

Explanation This program was included in the CCMACINC table and was dynamically attached. The program is translated to command level.

CCLL101I: *applid*: CICS COMET DYNAMIC ATTACH BYPASSED FOR: *progname*

Explanation The program *progname* has been excluded by the CCMACEXC table and is not dynamically attached. Unless this program is relinked with the CICS Comet stub, it is not translated to command level.

CCLL169I: *applid*: CICS COMET FOUND A MVS, ACCESS AT + *offset* IN *progrname*

Explanation Informational message produced when the program, *progname*, is dynamically attached by CICS Comet at hex *offset*.

CCLL191I: *applid*: CICS COMET FOUND A PRE-ESA RMI CALL AT *offset* IN: *progname*

Explanation CICS Comet did find an RMI call in this program. The FIX-RMI flag in the CCMACINC table was set to Y or left blank, indicating CICS Comet should search for RMI calls.
If a program's FIX-RMI flag is set to Y or left blank but it does not receive this message, you may change the FIX-RMI flag to N to speed Dynamic Attach processing.

CCLL192I: *_applid_*: CICS COMET FOUND A PRE-ESA CSA LOCATE AT *offset* IN: *progname*

Explanation

CICS Comet did find a CSA LOCATE command in this program. The FIX-CSA flag in the CCMACINC table was set to Y or left blank, indicating that CICS Comet should search for CSA LOCATE commands.
If a program's FIX-CSA flag is set to Y or left blank but it does not receive this message, you may change the FIX-CSA flag to N to speed Dynamic Attach processing.

CCLL193I: *_applid_*: CICS COMET FOUND A CICS/TS RMI CALL AT *offset* IN: *progname*

Explanation

CICS Comet did find an RMI call in this program. The FIX-RMI flag in the CCMACINC table was set to Y or left blank, indicating that CICS Comet should search for RMI calls.
If a program's FIX-RMI flag is set to Y or left blank but it does not receive this message, you may change the FIX-RMI flag to N to speed Dynamic Attach processing.

CCLL194I: *_applid_*: CICS COMET FOUND A CICS/TS CSA LOCATE AT *offset* IN: *progname*

Explanation

CICS Comet did find a CSA LOCATE command in this program. The FIX-CSA flag in the CCMACINC table was set to Y or left blank, indicating that CICS Comet should search for CSA LOCATE commands.

If a program's FIX-CSA flag is set to Y or left blank but it does not receive this message, you may change the FIX-CSA flag to N to speed Dynamic Attach processing.

CCLL196I: *_applid_*: CICS COMET FOUND A PRE-CICS/TS DFHCBLI AT +0000 IN *progname*

Explanation

CICS Comet found a DFHCBLI stub in this program. The FIX-CBL flag in the CCMACINC table was set to Y or left blank, indicating that CICS Comet should search for and fix the DFHCBLI stub.

CCLL198I: *_applid_*: CICS COMET FOUND A CICS/TS DFHCBLI AT +0000 IN *progname*

Explanation

This informational message lets you know that CICS Comet did find a DFHCBLI stub in this program. The FIX-CBL flag in the CCMACINC table was set to Y or left blank, indicating that CICS Comet should search for and fix the DFHCBLI stub.

CCLL199W: *_applid_*: CICS COMET FOUND: ???????? TO BE AN UNKNOWN FORMAT FOR A PL/I PROGRAM

Explanation

CICS Comet did not recognize the release of this PL/I program, or the program was link edited incorrectly. The program will abend with a CICS PLIE abend. Have a CICS transaction dump with program storage ready for viewing.

CCLL211E: *applid*: CICS COMET DETECTED ONE OR MORE REFRESH ERRORS

Explanation A NEWCOPY failed during a HALLEY REFRESH because either the program was in use or an attempt was made to dynamically attach CICS Comet exit routines. Verify the program is not in use and exclude CICS Comet exits from being attached.

CCLL218W: *applid*: PROGRAM WAS IN USE DURING NEWCOPY

Explanation A NEWCOPY failed during a HALLEY REFRESH because a program was in use.

CCLL666S: *applid*: CICS COMET MUST BE EXECUTED IN INITIALIZATION STAGE 3

Explanation CICS Comet started from the PLT before the `PROGRAM=DFHDELIM` entry. Place the CICS Comet entry after DFHDELIM, and restart CICS. Refer to “Install CICS Comet into Your CICS system” on page 10 for more information.

CCLL900E: *applid*: CICS COMET DETECTED AN INVALID LOADER CODE: *nn*

Explanation Contact UNICOM Systems with the loader code.

CCLL920S: *ddname* OPEN ERROR RC=(???) EC=(???) Explanation The DD statement for the CICS Comet log is missing in your JCL.

CCLL990W: CICS COMET ON CPU: *id* EXPIRES IN *nnn* DAYS

Explanation The CICS Comet password will expire in *nnn* days. Contact your UNICOM Systems account representative for a new password.

CCLL991W: CICS COMET LICENSE|TRIAL ON CPU: *id*

CCLL991W: REMAINING CICS COMET EXPIRED *xxx* DAYS AGO, YOU HAVE *nnn* DAYS

Explanation The CICS Comet password expired *xxx* days ago. You have *nnn* days left of a temporary grace period before CICS Comet becomes inactive. Contact UNICOM Systems account representative for a new password.

CCLL992E: *applid*: ALL PRODUCT CODES INVALID FOR THIS CPU

Explanation This message indicates you have attempted to start CICS Comet and the password has expired, or you are operating CICS Comet on an unlicensed CPU.

CCLL993S: CICS COMET PRODUCT CODE: *xxxx* INVALID FOR CPU *id*

Explanation The product code *xxxx* is invalid for CPU *id*. Check your password zaps. The product code is the four digit number following the REP keyword.

CCLL993S: CICS COMET PRODUCT CODE: *xxxx* EXPIRED FOR CPU *id*

Explanation The product code *xxxx* has expired for the CPU named *id*. The product code is the four digit number following the REP keyword in the password zaps. Contact your UNICOM Systems Customer Service for a new product code.

# Common Abend Codes

The following list describes CICS Comet and selected IBM abends. The IBM abends usually occur when CICS Comet is used incorrectly.

ABM*x* If you receive ABEND ABM0 or ABMx from your application programs, make sure your BMS MAPS are defined as “MAPSETS” in the PPT or through CEDA. Maps defined as assembler are regarded by Dynamic Attach as assembler programs and have the Comet command level stub prefixed to them if you generically include this program name in CCMACINC. You can exclude the BMS MAP in CCMACEXC if you included it generically in CCMACINC.

AEY8 This abend can occur when an application program contains a CICS Comet command level stub, but CICS Comet was never started. This happens most often when using the relink technique instead of using CICS Comet's Dynamic Attach.

APC6 This abend can occur when using VS/COBOL or COBOL II. If it occurs make sure you have the LE installed correctly.

APUZ This abend can occur when using CEDA to expand the CICS Comet CSD group. This abend usually occurs when the wrong Comet CSD deck has been used.

ASRD This abend occurs when an application program tries to access the CSA passed to it in R13 or by the DFHEIENT macro when using mixed-mode processing. This program should be placed under CICS Comet's control by placing an entry for it in the CCMACINC table. You must specify that this is a mixed-mode program by placing a Y or blank in the column marked MIXED. If you don't, the program fails with an ASRD abend.

AXFA A File Control Request sent to a remote system has a key length of zero in the FCT.

BADL CICS Comet detected a VS/COBOL command level program that was linked with an entry point of 0 instead of x’48’. Relink the program correctly.

C2ER A COBOL II error was detected. Make certain you have COBOL II generated in your COBOL II libraries and included in DFHRPL and STEPLIB.

C2RC A COBOL II error was detected. Make certain you have COBOL II generated in your COBOL II libraries and included in DFHRPL and STEPLIB.

CBL2 A COBOL II error was detected. Make certain you have COBOL II generated in your COBOL II libraries and included in DFHRPL and STEPLIB.

CC$E The assembler program is link edited with the command level stub, DFHE*, but its entry logic reflects a macro level program (using R14 for base).

CC$S The CICS Comet system TCA has been overlaid by your application. Have a transaction dump available when contacting Customer Service.

CC$T The USER TWA size is too small and has been overlaid. Increase the size of the TWA within PCT to the size required by the application transaction.

CC$U The USER TCA has been overlaid by your application. Have a transaction dump available when contacting Customer Service.

CC$X CICS Comet did not find the command level stub, but established addressability using Register 15 (command level). The application is link edited as a macro level program, but appears to have command level entry logic. Verify if the command level stub is at offset 0 in the load module.

The program may be improperly linked or it could indicate a G.L.U.E or a T.R.U.E, which are extensions to CICS and not supported by CICS Comet.

CC$Y The assembler program is link edited with the command level stub, DFHY*, but its entry logic reflects a macro level program(using R14 for base).

CCXY Equivalent to IBM's AEXY. This abend usually occurs during a short on storage condition. The executing transaction is no longer loaded. The transaction abends with a CICS transaction dump.

CCXZ Equivalent to IBM's AEXZ abend. There is a severe problem with the GETMAIN PARMLIST or with the storage domain. The transaction is abended with a CICS transaction dump.

NEIS This abend can occur when an application program contains a reference to the CSA, CWA, TCA, or TWA and is not running under CICS Comet's control. This can also be caused by mixed-mode programs not running under CICS Comet's control. Add an entry to CCMACINC and issue the `HALLEY REFRESH MACRO` command to put the mixed-mode program under CICS Comet's control.

☞ Even if a program makes no macro level calls but references the CSA or TCA, it must be considered a mixed-mode application program.

TCA@ An invalid assembler TCA address was passed to CICS Comet. Make certain all programs executed under this transaction are included in CCMACINC. If any are mixed-mode, remember to set the mixed-mode flag.

# Comet SAMPLIB Members

The following alphabetic list describes sample members distributed with this release of CICS Comet. These members are located in the SAMPLIB dataset created during CICS Comet

installation.

| | |
|---|---|
| $AMBLIST | Sample JCL for IBM's AMBLIST utility. |
| $COBOL | Sample zap for COBOL II WSCLEAR. |
| $$RELINK | Sample link edit JCL to correctly order CSECTs in a macro or command level CICS program. |
| ASMEPLNK | Sample assembly and relink of a ASM macro level program that does not have a standard entry point offset of 0. Not required for COBOL or PL/I. |
| ASMELINK | Sample assembly and relink of a ASM mixed-mode program that does not have a standard entry point offset of 0. Not required for COBOL or PL/I. |
| ASMMACRO | Sample relink of an ASM macro level program. |
| ASMMIXED | Sample relink of an ASM mixed-mode program. |
| BMSMAPS | Sample relink of BMS Maps to use them in XA storage. |
| CBLMACRO | Sample relink of a COBOL macro level program. |
| CBLMIXED | Sample relink of a COBOL macro and command level mixed-mode program. |
| CC$$V540 | Sample CICS/TS Dynamic Attach Initialization JCL. |
| CC$EXITS | List of all CICS Comet exits. |
| CCCMDEXC | Sample Exclude table of command level programs for CSA. This is the default member name only. |
| CCCMDINC | Sample Include table of command level programs for CSA. This is the default member name only. |
| CCEABMSP | Sample exit called after BMS maps are moved to XA storage. Full function. |
| CCEABMSX | Sample exit called after BMS maps are moved to XA storage. Fast function. |
| CCEAEEIP | Sample exit called after command level Include/Exclude tables are scanned. |
| CCEAEFCP | Sample exit called after each file control program type request. |
| CCEAEPCP | Sample exit called after each program control program type request. |
| CCEAKERN | Sample exit called after CICS Comet receives the converted command level response from CICS. |
| CCEACPEI | Sample exit called after CICS Comet copies the TCA/TWA to its own work area. |
| CCEBBMSP | Sample exit called before BMS maps are moved to XA storage. Full function |
| CCEBBMSX | Sample exit called before BMS maps are moved to XA storage. Fast function. |
| CCEBDATE | Sample exit called just before COMET refreshes the CSA date and time fields in the Comet CSA from the real CSA. |
| CCEBEEIP | Sample exit called before command level Include/Exclude tables are scanned. |
| CCEBEFCP | Sample exit called before each file control program type request. |

| | |
|---|---|
| CCEBEPCP | Sample exit called before each program control program type request. |
| CCEBKERN | Sample exit called before CICS Comet converts a macro level request to command level. |
| CCEBCPEI | Sample exit called before CICS Comet copies the TCA/TWA to its own work area. |
| CCMACEXC | Sample Exclude table of macro level programs. This is the default member name only. |
| CCMACINC | Sample Include table of macro level programs. This is the default member name only. |
| CCSIPARM | Sample start-up parameters and passwords. This is the default member name only. |
| DFHCSADS | List of supported CSA fields provided in CICS Comet's CSA. |
| DFHDCTCC | Sample non-XA/remote DCT entry. |
| DFHJCTCC | Sample non-XA JCT entry. |
| DFHFCTCC | Sample non-XA/remote FCT entry. |
| DFHPCTCC | Sample non-XA PCT entries. |
| DFHPLTCC | Sample non-XA PLT entries. |
| DFHPPTCC | Sample non-XA PPT entries. |
| DFHTCTCC | Sample non-XA TCT entry. |
| DFHTSTCC | Sample non-XA TST entry. |
| IBMBEOCA | Sample JCL to relink the IBM PL/I "On Error" handler. |
| IN25UEXI | Sample exit called by InterTest for BALR instructions. |
| IN25LETX | Sample exit called by InterTest for SHARED storage alterations. |
| INTRTEST | Sample PTFs for InterTest. While we have provided samples for your convenience, we recommend that you get the PTFs straight from CA. |
| IVPCMENU | Sample COBOL program to access CSA in CICS or CICS Comet. |
| PLIMACRO | Sample relink of a PL/I macro level program. |
| PLIMIXED | Sample relink of an PL/I macro and command level mixed-mode program. |
| SUPERZAP | Sample JCL that can be used to apply PTFs to CICS Comet. |
| UPGRADE | Sample JCL to upgrade CICS Comet's PPT/PCT to DFHCSD. |

# Sample COBOL Program to Acquire CSA Address

IVPCMENU is an example COBOL program that acquires the address of the CSA provided by CICS Comet and the address of the real CSA.

```
*****************************************************************
* IVPCMENU - SAMPLE COBOL PROGRAM FOR ACQUIRING CSA ADDRESS
*****************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. IVPCMENU.
 ENVIRONMENT DIVISION.
*
DATA DIVISION.
 WORKING-STORAGE SECTION.
   77 WSEYEBALL  PIC X(24) VALUE 'WORKING STORAGE SECTION'.
   01 COMMAREA SYNCHRONIZED.
   02 FUNCTION PICTURE XXXX.
   02 BLLCELL PICTURE S9(8) COMP.
*
 LINKAGE SECTION.
 01   DFHBLLDS SYNCHRONIZED.
    02    BLLCBAR          PICTURE S9(8) COMP.
    02    CSACBAR          PICTURE S9(8) COMP.
    02    CSAOPBAR         PICTURE S9(8) COMP.
    02    TCACBAR          PICTURE S9(8) COMP.
*
          COPY DFHCSADS.
          COPY DFHTCADS.
*
 PROCEDURE DIVISION.
*****************************************************************
* GET CICS COMET CSA ADDRESS FIRST
*****************************************************************
          MOVE 'CCSA' TO FUNCTION.
          EXEC CICS LINK PROGRAM('CC$$CCSA')
                    COMMAREA(COMMAREA)
                    LENGTH(8)
           END-EXEC.
           MOVE BLLCELL TO CSACBAR.
           SERVICE RELOAD DFHCSADS.
           MOVE CSACDTA TO TCACBAR.
           SERVICE RELOAD DFHTCADS.
*****************************************************************
* GET REAL CSA ADDRESS NEXT
*****************************************************************
          MOVE 'REAL' TO FUNCTION.
          EXEC CICS LINK PRORGAM('CC$$CCSA')
```

```
                COMMAREA(COMMAREA)
                LENGTH(8)
        END-EXEC.
*****************************************************************
* SEND BMS MAP TO SCREEN
*****************************************************************
        EXEC CICS SEND MAP('MENU')
                MAPSET('DFH$CGA')
                MAPONLY ERASE
        END-EXEC.
        EXEC CICS RETURN
        END-EXEC.
GOBACK.
```

# Appendix B. Commands and Debugging

This appendix describes CICS Coment commands and diagnostic procedures to debug problems.

## CICS Comet Commands

The following list is an alphabetic summary of CICS Comet commands. Page references are included for further information about the command.

HALLEY (blank) — Display CICS Comet system wide statistics. (You must enter a space after HALLEY. Refer to page 113 for instructions to display the CICS Comet version number.

HALLEY DEBUG — Toggle the RMODE flag in the Comet Include table for debuggers expecting 24-bit VS/COBOL programs. The command format is:

HALLEY DEBUG=program

where program is an exact match of an Include table entry, including wildcards. If necessary, use blanks to pad program to exactly 8-characters in length.

☞ After you issue the DEBUG command, you must manually NEWCOPY the affected programs.

HALLEY COMET — Verify installation of CICS Comet. Refer to page 24 for instructions to verify that CICS Comet has been installed correctly.

HALLEY INQUIRY — Display CICS Comet detail statistics.

HALLEY LIST EXCLUDED COMMAND PROGRAMS or HALL L E C

Display a list of all the command level programs that CICS Comet excludes from the Include table. The programs appear in the order that they are entered in the CCCMDEXC member of the SAMPLIB dataset. Refer to page 39 for instructions to display command level programs entered in the CICS Comet Include and Exclude tables.

HALLEY LIST EXCLUDED MACRO PROGRAMS or HALL L E M

Display a list of all the macro level programs that CICS Comet excludes from the Include table. The macro level programs are listed in the order that they are entered in the CCMACEXEC member of the SAMPLIB dataset.

HALLEY LIST INCLUDED COMMAND PROGRAMS or HALL L I C

Display a list of all the command level programs for which CICS Comet will use its CSA. CWA, or TWA. The list is displayed in the order that they are entered in the CCCMDINC member of the SAMPLIB dataset. Refer to page 39 for instructions to display command level programs entered in the CICS Comet Include and Exclude tables.

HALLEY LIST INCLUDED MACRO PROGRAMS or HALL L I M

Display a list of all the macro level programs that can be dynamically attached to CICS Comet. Macro programs are displayed in the order that they are entered in the CCMACINC member of the SAMPLIB dataset. Refer to page 17 for instructions to list macro level programs.

HALLEY REFRESH COMMAND [PROGRAMS]

Reload the list of all the command level programs for which CICS Comet will create a special CSA. The list is loaded from the CCCMDINC member of the SAMPLIB dataset. Refer to page 99 for more information.

HALLEY REFRESH DATASETS

Refresh the CICS Comet DFHFCTCC table. Refer to page 100 for more information about using this command.

HALLEY REFRESH MACRO [PROGRAMS]

Reload the list of all the macro level programs that are dynamically attached to CICS Comet. The list is loaded from the CCMACINC member of the SAMPLIB dataset. Refer to page 99 for more information.

HALLEY REFRESH JOURNALS

Refresh the CICS Comet DFHJCTCC table.

HALLEY REFRESH *domain*

Refresh the CICS Comet domain. The refresh does a release and NEWCOPY and loads *domain* and both exits related to this domain

HALLEY REFRESH PROGRAMS

Refresh the CICS Comet DFHPPTCC table. Refer to page 74 for instructions to refresh the DFHPPTCC table

HALLEY REFRESH TEMPSTOR

Refresh the CICS Comet DFHTSTCC table. Refer to page 76 for instructions to refresh the DFHTSTCC table.

HALLEY REFRESH TERMINALS

Refresh the CICS Comet DFHTCTCC table. Refer to page 75 for instructions to refresh the DFHTCTCC table.

HALLEY REFRESH TRANSACTIONS

Refresh the CICS Comet DFHPCTCC table. Refer to page 73 for instructions to refresh the DFHPCTCC table.

HALLEY REFRESH TRANSDATA

Refresh the CICS Comet DFHDCTCC table. Refer to page 71 for instructions to refresh the DFHDCTCC table.

HALLEY RESET — Reset all CICS Comet detail statistic counters to zero. Refer to page 116 for instructions to reset detail statistics counters.

HALLEY SNAPDUMP

Produces a snap dump that is placed in the CICS dump dataset. Refer to page 44 for instructions to use this command to collect diagnostic data for UNICOM Systems Customer Service.

HALLEY START — Start CICS Comet from any CICS terminal. Refer to page 23 for instructions to start CICS Comet.

HALLEY TRACE=ON

Turn on the Comet extended macro statement trace facility.

HALLEY TRACE=OFF

Turn off the Comet extended macro statement trace facility.

HALLEY VERSION

Displays the version of CICS Comet. Refer to page 44 for instructions to use this command to collect diagnostic data for UNICOM Systems Customer Service

# HALY Service Transactions

This section gives a summary of CICS Comet service transactions.

HALY LOCATE COMETCSA

Finds and displays the CICS Comet CSA in storage.

HALY LOCATE COMETOFL

Finds and displays CICS Comet CSA optional feature list storage.

HALY LOCATE COMETSIT

Finds and displays CICS Comet SIT storage.

HALY LOCATE COMETCWA

Finds and displays the CICS Comet CWA in storage.

HALY LOCATE DFHCSA

Finds and displays the CICS CSA in storage.

HALY LOCATE DFHPPT *program*

Displays PPT entry for this *program*.

HALY LOCATE *program*

Displays storage for *program,* if currently loaded into DSA program storage.

HALY PROGRAM *program*

Displays the program attribute screen for *program*. You can also NEWCOPY or LOAD a program by placing the letter N or L in the first byte of the program status field and pressing ENTER. The status field is normally ENA or DIS, for enabled or disabled.

☞ The command LOCATE can be abbreviated to L, and PROGRAM to PROG.

### HALLEY REFRESH Macro Programs (CC$$NEWC)

This command reloads the macro level Include and Exclude tables from the PDS dataset pointed to by the CCMACINC and CCMACEXC DD statements. The tables are kept in extended storage and are anchored from the CC$$ESIP module.

After the Include and Exclude tables have been reloaded, the refresh command links to the automatic CC$$NEWC newcopy program. In CICS version 2.1, CC$$NEWC sequentially scans PPT table entries looking for programs that should have the CICS Comet stub in storage but don't, and for programs that have the CICS Comet stub in storage but shouldn't. If either case is detected and the program is in storage, an `EXEC CICS NEWCOPY` command is issued to refresh the program in storage.

In CICS/TS and above, CC$$NEWC sequentially scans the CPE table entries, looking for programs that should have the CICS Comet stub but don't, or programs that have the CICS Comet stub but shouldn't.

### HALLEY REFRESH COMMAND Programs

This command reloads the command level Include and Exclude tables from the PDS member pointed to by the CCCMDINC and CCCMDEXC DD statements. The tables are kept in extended storage and are anchored from the CC$$EEIP module.

### HALLEY REFRESH CC$$E???

This command gives CICS Comet the ability to newcopy a CICS Comet domain (except for the kernel domain, CC$$MAIN) and the exits associated with the domain, after zaps have been applied or a new exit has been reassembled.

Issue this command only in a quiesced environment because the program associated with the command is deleted from storage and reloaded from the PDS load library. Anyone using the program at that time the command is entered may abend with a S0C1/ASRA.

You can use this command to immediately activate an exit. To do so:

1. Add the PPT entry needed for the CICS Comet exit.
2. Load the exit into storage with CECI LOAD PROGRAM ('program') HOLD.
3. Issue the REFRESH CC$$E??? command. The exit is now activated.

Also, an exit can be deactivated by disabling the exit's PPT entry and then issuing the `HALLEY REFRESH CC$$E???` command.

### HALLEY REFRESH Datasets

REFRESH DATASETS works only if you have coded a dummy FCT table and specified the suffix in CCSIPARM. For example, you specified FCT=CC and issued the REFRESH DATASETS command, CICS Comet will newcopy the DFHFCTCC module in the PPT and then scan through the dummy FCT, dataset by dataset, and issue an EXEC CICS INQUIRE DATASET(dataset) to get the dataset's attributes.

The attributes are then reformatted into options that can be stored in the dummy FCT entries. This gives Cometized application programs the ability to scan through CICS FCT control blocks, as they did in CICS 1.7 and 2.1.

Refer to page 96 for a summary of similar refresh commands for programs, transient data, transactions, and journals.

### HALY PROGRAM

HALY PROGRAM is a service transaction to display program attributes such as language, size, entry offsets, library number, etc.

The following screen appears after entering HALY PROGRAM. You can view a program's PPT entry in storage by positioning the cursor over the program name line and pressing PF2. Press PF3 to return to the previous screen.

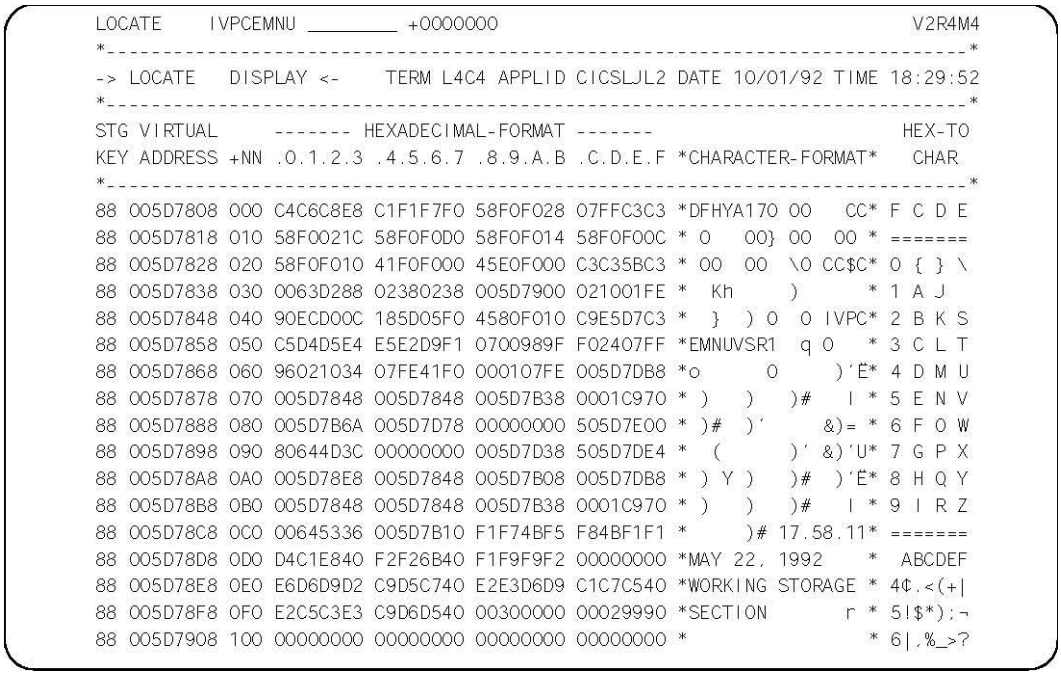
```
PROGRAM   IVPCEMNU                                                          V2R4M4
*-------------------------------------------------------------------------------*
-> PROGRAM  DISPLAY <-    TERM L4C4 APPLID CICSLJL2 DATE 10/01/92 TIME 18:23:28
*-------------------------------------------------------------------------------*
PROGRAM    USE   CUR FETCH -STATUS- PROG  PROGRAM ENTRY  PROGRAM -DASD- LIB
  ID.     COUNT  USE COUNT RES. E/D LANG  ADDRESS OFFSET LENGTH  T.T.R. NUM
*-------------------------------------------------------------------------------*
IVPCEMNU ______1 __0 ____1 CORE ENA ASM  005D7808 000010 ___3912 000307 _10
IVPC2ALL ______0 __0 ____0 NO   ENA COBL
IVPC2BRW ______0 __0 ____0 NO   ENA COBL
IVPC2GA  ______0 __0 ____0 NO   ENA ASM
IVPC2GB  ______0 __0 ____0 NO   ENA ASM
IVPC2GC  ______0 __0 ____0 NO   ENA ASM
IVPC2MNU ______0 __0 ____0 NO   ENA COBL
IVPC2S2K ______0 __0 ____0 NO   ENA COBL
IVPP$ALL ______0 __0 ____0 STG  ENA PL/1 006B8940        __13712 00030B _13
IVPP$BRW ______0 __0 ____0 STG  ENA PL/1 006BBED0        __13128 000324 _13
IVPP$GA  ______0 __0 ____1 OSLD ENA ASM  82D0F438        ____512 FFFFFF 256
IVPP$GB  ______0 __0 ____1 OSLD ENA ASM  82D0F278        ____448 FFFFFF 256
IVPP$GC  ______0 __0 ____1 OSLD ENA ASM  82D0F088        ____496 FFFFFF 256
IVPP$MNU ______0 __0 ____0 STG  ENA PL/1 006BF220        ____800 00022E _13
IVPPEALL ______0 __0 ____0 NO   ENA PL/1
IVPPEBRW ______0 __0 ____0 NO   ENA PL/1
IVPPEMNU ______0 __0 ____0 NO   ENA PL/1
```

**HALY LOCATE Command**

The HALY LOCATE command is a service transaction to display the contents of program storage. Displaying program storage allows you to examine the beginning of the program in memory and verify if the CICS Comet stub or a command level stub is present.

HALY LOCATE can also find many CICS management modules and control blocks such as: DFHCSA, DFHTCA, DFHPPT, DFHPCT, DFHFCT, DFHTCT, DFHDCT. Control blocks (like the FCT) can be entered as a file name to find a particular FCT entry instead of the start of the control blocks.

For example: HALY LOCATE DFHFCT filename.




Press PF3 to return to the PROGRAM screen after you have located the program.

**HALY LOCATE DFHPPT Command**

HALY LOCATE DFHPPT is a service transaction to display the program's PPT entry in virtual storage. Displaying program PPT storage allows you to examine program settings in the PPT and verify if the CICS Comet stub or a command level stub is present.

LOCATE can also find many CICS management modules and control blocks such as: DFHCSA, DFHTCA, DFHPPT, DFHPCT, DFHFCT, DFHTCT, DFHDCT. Control blocks like the FCT can be entered as a file name to find a particular FCT entry instead of the start of the control block.

For example: HALY LOCATE DFHFCT filename.

```
LOCATE    DFHPPT   IVPCEMNU +0000000                                        V2R4M4
*-------------------------------------------------------------------------------*
-> LOCATE   DISPLAY <-    TERM L4C4 APPLID CICSLJL2 DATE 10/01/92 TIME 18:28:13
*-------------------------------------------------------------------------------*
STG VIRTUAL     ------- HEXADECIMAL-FORMAT -------                        HEX-TO
KEY ADDRESS +NN .0.1.2.3 .4.5.6.7 .8.9.A.B .C.D.E.F *CHARACTER-FORMAT*   CHAR
*-------------------------------------------------------------------------------*
88 0002F574 000 C9E5D7C3 C5D4D5E4 00030709 005D7808 *IVPCEMNU     )  * F C D E
88 0002F584 010 00000F48 00000010 00280000 66808000 *                 * =======
88 0002F594 020 000C0001 00000001 00000001 93000040 *            I    * 0 { } \
88 0002F5A4 030 C9E5D7C3 F2C1D3D3 00000000 00000000 *IVPC2ALL         * 1 A J
88 0002F5B4 040 00000000 00000000 00340010 60008000 *            -    * 2 B K S
88 0002F5C4 050 000C0000 00000000 00000000 00000000 *                 * 3 C L T
88 0002F5D4 060 00000000 00000000 00000000 93000040 *            I    * 4 D M U
88 0002F5E4 070 C9E5D7C3 F2C2D9E6 00000000 00000000 *IVPC2BRW         * 5 E N V
88 0002F5F4 080 00000000 00000000 00340010 60008000 *            -    * 6 F O W
88 0002F604 090 000C0000 00000000 00000000 00000000 *                 * 7 G P X
88 0002F614 0A0 00000000 00000000 00000000 93000030 *            I    * 8 H Q Y
88 0002F624 0B0 C9E5D7C3 F2C7C140 00000000 00000000 *IVPC2GA         * 9 I R Z
88 0002F634 0C0 00000000 00000000 00280000 60004000 *            -    * =======
88 0002F644 0D0 000C0000 00000000 00000000 93000030 *            I    *  ABCDEF
88 0002F654 0E0 C9E5D7C3 F2C7C240 00000000 00000000 *IVPC2GB         * 4¢.<(+|
88 0002F664 0F0 00000000 00000000 00280000 60004000 *            -    * 5!$*);¬
88 0002F674 100 000C0000 00000000 00000000 93000030 *            I    * 6|,%_>?
```

Press PF3 to return to the PROGRAM screen after you have located the program.

**HALY LOCATE COMETCSA (points to CICS Comet CSA)**

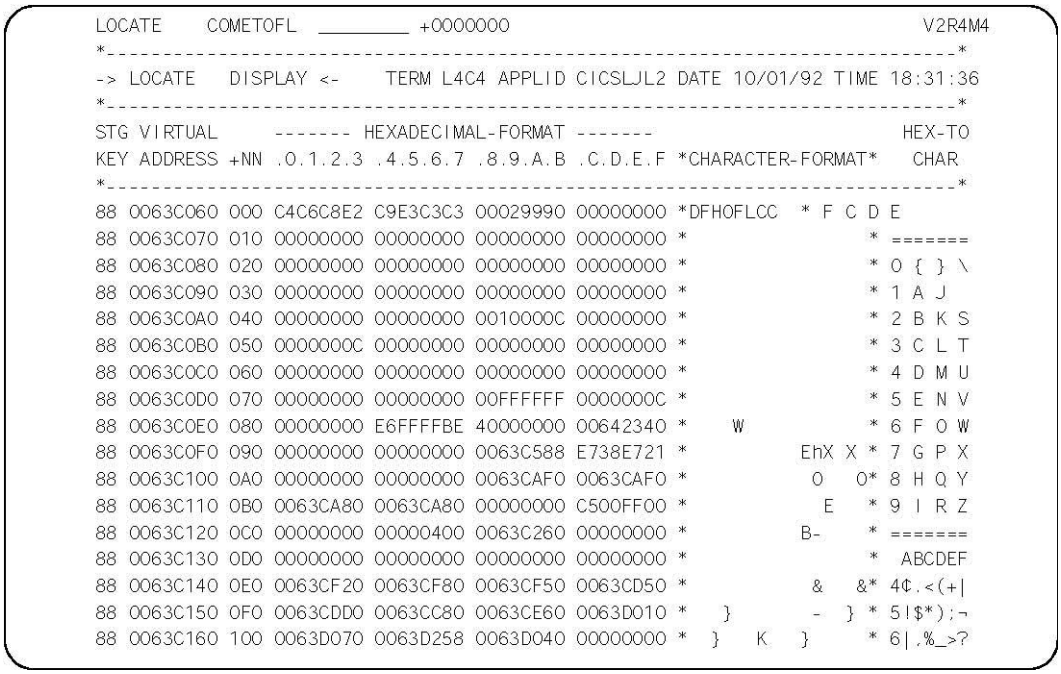
A special form of the LOCATE command can be used to find the CICS Comet copy of the CSA. You can locate CICS Comet CSA storage by entering HALY LOCATE COMETCSA. You can verify this is the CICS Comet CSA by pressing PF7 (scroll backward) and finding the “CCDFHCSA” eye-catcher.

The CICS Comet CWA follows the CSA by 512 bytes or X’200‘. To display the contents of the CWA, enter in an offset of +0000200 on the command line and press ENTER, or enter HALY LOCATE COMETCWA.

```
LOCATE    COMETCSA  ________ +0000000                                        V2R4M4
*-------------------------------------------------------------------------------*
-> LOCATE   DISPLAY <-    TERM L4C4 APPLID CICSLJL2 DATE 10/01/92 TIME 18:31:36
*-------------------------------------------------------------------------------*
STG VIRTUAL     ------- HEXADECIMAL-FORMAT -------                        HEX-TO
KEY ADDRESS +NN .0.1.2.3 .4.5.6.7 .8.9.A.B .C.D.E.F *CHARACTER-FORMAT*   CHAR
*-------------------------------------------------------------------------------*
88 0063C060 000 C3C3C4C6 C8C3E2C1 00029990 00000000 *CCDFHCSA  r     * F C D E
88 0063C070 010 00000000 00000000 00000000 00000000 *                 * =======
88 0063C080 020 00000000 00000000 00000000 00000000 *                 * 0 { } \
88 0063C090 030 00000000 00000000 00000000 00000000 *                 * 1 A J
88 0063C0A0 040 00000000 00000000 0010000C 00000000 *                 * 2 B K S
88 0063C0B0 050 0000000C 00000000 00000000 00000000 *                 * 3 C L T
88 0063C0C0 060 00000000 00000000 00000000 00000000 *                 * 4 D M U
88 0063C0D0 070 00000000 00000000 00FFFFFF 0000000C *                 * 5 E N V
88 0063C0E0 080 00000000 E6FFFFBE 40000000 00642340 *    W            * 6 F O W
88 0063C0F0 090 00000000 00000000 0063C588 E738E721 *           EhX X * 7 G P X
88 0063C100 0A0 00000000 00000000 0063CAF0 0063CAF0 *            0   0* 8 H Q Y
88 0063C110 0B0 0063CA80 0063CA80 00000000 C500FF00 *              E   * 9 I R Z
88 0063C120 0C0 00000000 00000400 0063C260 00000000 *           B-     * =======
88 0063C130 0D0 00000000 00000000 00000000 00000000 *                  *  ABCDEF
88 0063C140 0E0 0063CF20 0063CF80 0063CF50 0063CD50 *            &   &* 4¢.<(+|
88 0063C150 0F0 0063CDD0 0063CC80 0063CE60 0063D010 *   }        -  } * 5!$*);¬
88 0063C160 100 0063D070 0063D258 0063D040 00000000 *  }   K   }     * 6|.%_>?
```

**HALY LOCATE COMETOFL (points to CICS Comet Optional Features List)**

Enter HALY LOCATE COMETOFL to find the CICS Comet copy of the Optional Features List.



```
LOCATE    COMETOFL  ________ +0000000                                    V2R4M4
*-------------------------------------------------------------------------*
-> LOCATE   DISPLAY <-    TERM L4C4 APPLID CICSLJL2 DATE 10/01/92 TIME 18:31:36
*-------------------------------------------------------------------------*
STG VIRTUAL     ------- HEXADECIMAL-FORMAT -------                    HEX-TO
KEY ADDRESS +NN .0.1.2.3 .4.5.6.7 .8.9.A.B .C.D.E.F *CHARACTER-FORMAT*   CHAR
*-------------------------------------------------------------------------*
88 0063C060 000 C4C6C8E2 C9E3C3C3 00029990 00000000 *DFHOFLCC  * F C D E
88 0063C070 010 00000000 00000000 00000000 00000000 *                * =======
88 0063C080 020 00000000 00000000 00000000 00000000 *                * 0 { } \
88 0063C090 030 00000000 00000000 00000000 00000000 *                * 1 A J
88 0063C0A0 040 00000000 00000000 0010000C 00000000 *                * 2 B K S
88 0063C0B0 050 0000000C 00000000 00000000 00000000 *                * 3 C L T
88 0063C0C0 060 00000000 00000000 00000000 00000000 *                * 4 D M U
88 0063C0D0 070 00000000 00000000 00FFFFFF 0000000C *                * 5 E N V
88 0063C0E0 080 00000000 E6FFFFBE 40000000 00642340 *    W           * 6 F O W
88 0063C0F0 090 00000000 00000000 0063C588 E738E721 *          EhX X * 7 G P X
88 0063C100 0A0 00000000 00000000 0063CAF0 0063CAF0 *           0   0* 8 H Q Y
88 0063C110 0B0 0063CA80 0063CA80 00000000 C500FF00 *             E   * 9 I R Z
88 0063C120 0C0 00000000 00000400 0063C260 00000000 *          B-     * =======
88 0063C130 0D0 00000000 00000000 00000000 00000000 *                 *  ABCDEF
88 0063C140 0E0 0063CF20 0063CF80 0063CF50 0063CD50 *           &   &* 4¢.<(+|
88 0063C150 0F0 0063CDD0 0063CC80 0063CE60 0063D010 *   }       -  } * 5!$*);¬
88 0063C160 100 0063D070 0063D258 0063D040 00000000 *  }   K   }     * 6|,%_>?
```

**HALY LOCATE COMETSIT (points to CICS Comet SIT)**

To locate the CICS Comet SIT storage enter the HALY LOCATE COMETSIT command.

```
LOCATE    COMETSIT  ________ +0000000                                    V2R4M4
*-------------------------------------------------------------------------------*
-> LOCATE   DISPLAY <-    TERM L4C4 APPLID CICSLJL2 DATE 10/01/92 TIME 18:31:36
*-------------------------------------------------------------------------------*
STG VIRTUAL     ------- HEXADECIMAL-FORMAT -------                      HEX-TO
KEY ADDRESS +NN .0.1.2.3 .4.5.6.7 .8.9.A.B .C.D.E.F *CHARACTER-FORMAT*   CHAR
*-------------------------------------------------------------------------------*
88 0063C060 000 C4C6C8E2 C9E3C3C3 00029990 00000000 *DFHSITCC  * F C D E
88 0063C070 010 00000000 00000000 00000000 00000000 *                * =======
88 0063C080 020 00000000 00000000 00000000 00000000 *                * 0 { } \
88 0063C090 030 00000000 00000000 00000000 00000000 *                * 1 A J
88 0063C0A0 040 00000000 00000000 0010000C 00000000 *                * 2 B K S
88 0063C0B0 050 0000000C 00000000 00000000 00000000 *                * 3 C L T
88 0063C0C0 060 00000000 00000000 00000000 00000000 *                * 4 D M U
88 0063C0D0 070 00000000 00000000 00FFFFFF 0000000C *                * 5 E N V
88 0063C0E0 080 00000000 E6FFFFBE 40000000 00642340 *    W           * 6 F O W
88 0063C0F0 090 00000000 00000000 0063C588 E738E721 *          EhX X * 7 G P X
88 0063C100 0A0 00000000 00000000 0063CAF0 0063CAF0 *           0   0* 8 H Q Y
88 0063C110 0B0 0063CA80 0063CA80 00000000 C500FF00 *             E   * 9 I R Z
88 0063C120 0C0 00000000 00000400 0063C260 00000000 *          B-     * =======
88 0063C130 0D0 00000000 00000000 00000000 00000000 *                 *  ABCDEF
88 0063C140 0E0 0063CF20 0063CF80 0063CF50 0063CD50 *           &   &* 4¢.<(+|
88 0063C150 0F0 0063CDD0 0063CC80 0063CE60 0063D010 *   }       -  } * 5!$*);¬
88 0063C160 100 0063D070 0063D258 0063D040 00000000 *  }   K   }     * 6|.%_>?
```

**CICS Comet's EIS Trace Table**

**Finding the CICS Comet EIS Area**

Complete the following procedure to view the contents of CICS Comet EIS storage:

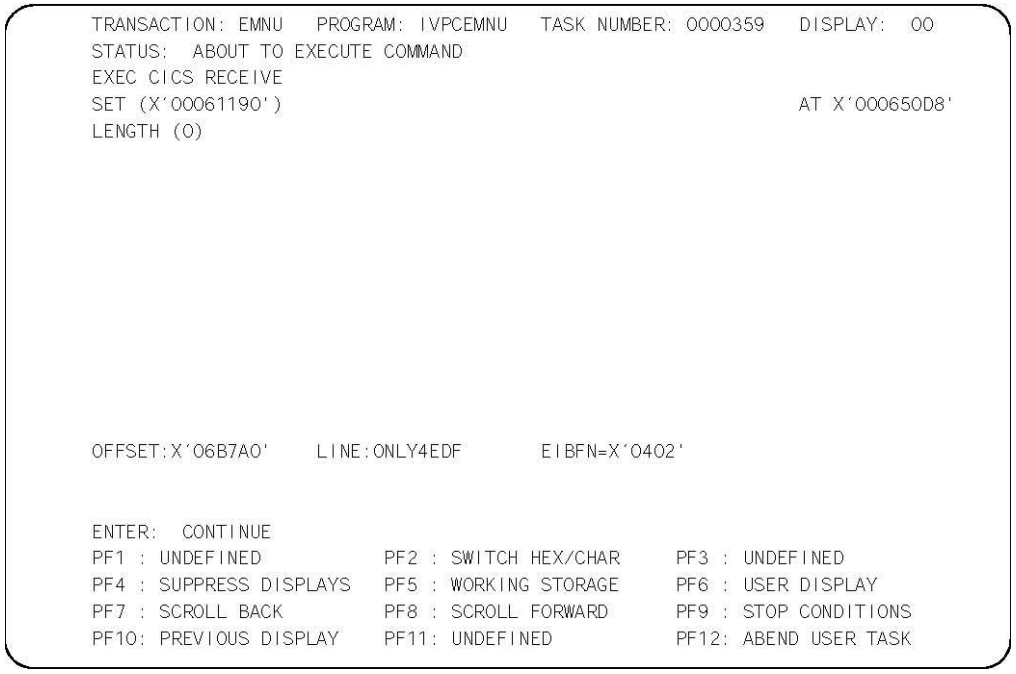
1. Use the CEDF transaction and find the CC$$EISS or CC$$EISM eye-catcher at offset x'0F8' in the storage area.

   The first EIS storage area is chained from the real CICS TCA at offset x'0EC' in CICS 2.1 and x'0E0' in CICS/TS and above.

   If the first area is not the CICS Comet EIS area, then subtract 4 bytes from your EIS address. At that location there will be a address starting at offset +0.

2. Enter this address in the CEDF screen to display the contents of storage.
3. Check for the CC$$EISx eye-catcher.

   If is not there, repeat the process until you find it, or you reach an address of 00000000 for the previous address (which means there is no CICS Comet EIS area).




In the screen shown above, you can see the real CICS TCA address in the field labeled `SET(x'????????')` if the LINE: ONLY4EDF appears on the CEDF screen.

This EXEC CICS RECEIVE screen line appears only when CICS Comet runs under CEDF. This allows the macro program to see the real TIOA, not the last CEDF screen.

Note: Before execution of this RECEIVE, CICS Comet places the real CICS TCA address into the SET(X'00000000') field.

In the screen shown below, you can see a Cometized macro statement being executed as its equivalent command level call. You can press PF5 to see the working storage area and then

enter the TCA address from the previous CEDF screen.

```
TRANSACTION: EMNU   PROGRAM: IVPCEMNU   TASK NUMBER: 0000359   DISPLAY:  00
STATUS:  ABOUT TO EXECUTE COMMAND
EXEC CICS SEND MAP
MAP ('MENU   ')
MAPONLY
LENGTH (0)
MAPSET ('IVPC$GA')
TERMINAL
ERASE
NOHANDLE




OFFSET:X'748B5E'    LINE:505D7CF8       EIBFN=X'1804'


ENTER:  CONTINUE
PF1 : UNDEFINED           PF2 : SWITCH HEX/CHAR     PF3 : UNDEFINED
PF4 : SUPPRESS DISPLAYS   PF5 : WORKING STORAGE     PF6 : USER DISPLAY
PF7 : SCROLL BACK         PF8 : SCROLL FORWARD      PF9 : STOP CONDITIONS
PF10: PREVIOUS DISPLAY    PF11: UNDEFINED           PF12: ABEND USER TASK
```

Note how the line number in the CEDF screen is the macro level statement's return address in your program.

In the screen below you can see the REAL CICS TCA area. The first EIS storage area pointer is at offset +0EC into the TCA area for CICS 2.1 (or offset +0E0 in CICS/TS).

```
TRANSACTION: EMNU   PROGRAM: IVPCEMNU   TASK NUMBER: 0000359   DISPLAY:  00
ADDRESS: 00061190
00061190   000000   00061000 00000000 00000000 007F6A00   ............"..
000611A0   000010   000647E0 01048E04 10040100 04810000   ............a..
000611B0   000020   405D6F00 40C9864C 00000028 00061014    )?. If<........
000611C0   000030   005D682A 0001CBD0 0002F574 00061000   .)........5.....
000611D0   000040   00049A90 00000000 007F4E54 00048E04   .........+.....
000611E0   000050   00000000 00064520 50C9A2F8 00049990   ........&Is8..r.
000611F0   000060   40C99374 00061000 0000B150 00001238    Il........&....
00061200   000070   007F81C8 0001C310 40C98652 00C99652   ."aH..C. If..Io.
00061210   000080   00000004 00004000 C4C6E700 82D1C1F8   ......  .DFX.bJA8
00061220   000090   D4C5D5E4 40404040 C9E5D7C3 5BC7C140   MENU    IVPC$GA
00061230   0000A0   005D6828 807F5BD2 80C9D8E8 00045DC8   .)..."$K.IQY..)H
00061240   0000B0   000650FC 00045E4D 000650DC 00045E56   ..&...;(..&...;.
00061250   0000C0   000650DC 00000004 007F4E54 007F5B28   ..&.....+..".$.
00061260   0000D0   00000000 00064520 00000000 00000000   ................
00061270   0000E0   00000000 00000000 04000000 00064FF0   ..............|0
00061280   0000F0   00000000 00000000 00000000 00000000   ................

ENTER:  CURRENT DISPLAY
PF1 : UNDEFINED           PF2 : BROWSE TEMP STORAGE PF3 : UNDEFINED
PF4 : EIB DISPLAY         PF5 : WORKING STORAGE     PF6 : USER DISPLAY
PF7 : SCROLL BACK HALF    PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL    PF11: SCROLL FORWARD FULL PF12: REMEMBER DISPLAY
```

Enter address 64FFO to get the following screen, containing the CICS Comet EIS storage area.

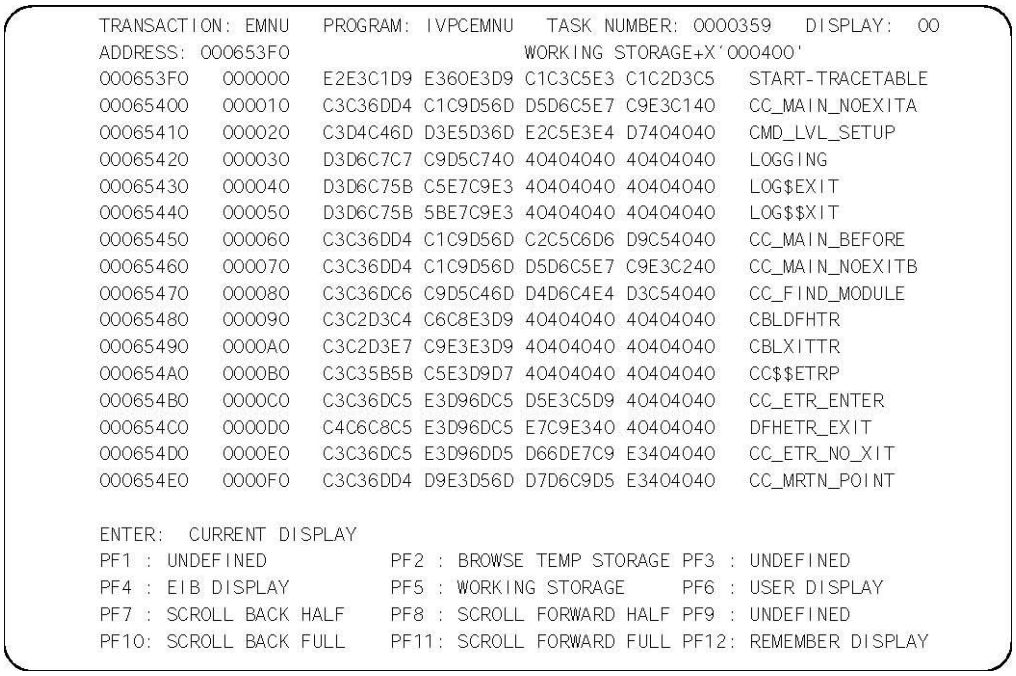
```
TRANSACTION: EMNU   PROGRAM: IVPCEMNU   TASK NUMBER: 0000359   DISPLAY:  00
ADDRESS: 00064FF0                      WORKING STORAGE
00064FF0   000000   000645E4 00029990 005D7900 82D20390   ...U..r..)..bK..
00065000   000010   00000000 00000000 00065058 000650FC   .........&...&.
00065010   000020   00000000 005D7D38 0063C260 00065BA0   .....)'...B-..$.
00065020   000030   00064688 00029990 0001C970 02D1FDB0   ...h..r...I..J..
00065030   000040   02D20DB0 00065EF0 00000000 00000000   .K....;0........
00065040   000050   00000000 007F6600 00029990 00064688   .....".....r....h
00065050   000060   80000000 00000000 000650FC 00065F80   ..........&...¬.
00065060   000070   000650DC 000650E2 00065F88 00000000   ..&...&S..¬h....
00065070   000080   00000000 00000000 00000000 00000000   ................
00065080   000090   00000000 00000000 00000000 00000000   ................
00065090   0000A0   00000000 80000000 00000000 00000000   ................
000650A0   0000B0   00000000 00000000 0C02E000 490000CC   ................
000650B0   0000C0   00F0F1F0 F4F8C5F0 F4000000 00000000   .01048E04.......
000650C0   0000D0   00000000 00000000 00000000 00000000   ................
000650D0   0000E0   00000000 00000000 00065EF0 00000000   ..........;0....
000650E0   0000F0   00060000 0001C970 C3C35B5B C5C9E2D4   ......I.CC$$EISM

ENTER:  CURRENT DISPLAY
PF1 : UNDEFINED           PF2 : BROWSE TEMP STORAGE PF3 : UNDEFINED
PF4 : EIB DISPLAY         PF5 : WORKING STORAGE     PF6 : USER DISPLAY
PF7 : SCROLL BACK HALF    PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL    PF11: SCROLL FORWARD FULL PF12: REMEMBER DISPLAY
```

**Locating the CICS Comet Trace Table**

1. Follow the procedure outlined in “CICS Comet’s EIS Trace Table” on page 106, to find the CICS Comet EIS area

2. Scroll forward using PF11 or PF8 until you find the eye-catcher “START-OF-TRACE” or “TRACETBL-START”.

The oldest entry is at the top of the trace table. More recent entries are towards the bottom.



```
TRANSACTION: EMNU   PROGRAM: IVPCEMNU   TASK NUMBER: 0000359   DISPLAY:  00
ADDRESS: 000653F0                    WORKING STORAGE+X'000400'
000653F0   000000   E2E3C1D9 E360E3D9 C1C3C5E3 C1C2D3C5   START-TRACETABLE
00065400   000010   C3C36DD4 C1C9D56D D5D6C5E7 C9E3C140   CC_MAIN_NOEXITA
00065410   000020   C3D4C46D D3E5D36D E2C5E3E4 D7404040   CMD_LVL_SETUP
00065420   000030   D3D6C7C7 C9D5C740 40404040 40404040   LOGGING
00065430   000040   D3D6C75B C5E7C9E3 40404040 40404040   LOG$EXIT
00065440   000050   D3D6C75B 5BE7C9E3 40404040 40404040   LOG$$XIT
00065450   000060   C3C36DD4 C1C9D56D C2C5C6D6 D9C54040   CC_MAIN_BEFORE
00065460   000070   C3C36DD4 C1C9D56D D5D6C5E7 C9E3C240   CC_MAIN_NOEXITB
00065470   000080   C3C36DC6 C9D5C46D D4D6C4E4 D3C54040   CC_FIND_MODULE
00065480   000090   C3C2D3C4 C6C8E3D9 40404040 40404040   CBLDFHTR
00065490   0000A0   C3C2D3E7 C9E3E3D9 40404040 40404040   CBLXITTR
000654A0   0000B0   C3C35B5B C5E3D9D7 40404040 40404040   CC$$ETRP
000654B0   0000C0   C3C36DC5 E3D96DC5 D5E3C5D9 40404040   CC_ETR_ENTER
000654C0   0000D0   C4C6C8C5 E3D96DC5 E7C9E340 40404040   DFHETR_EXIT
000654D0   0000E0   C3C36DC5 E3D96DD5 D66DE7C9 E3404040   CC_ETR_NO_XIT
000654E0   0000F0   C3C36DD4 D9E3D56D D7D6C9D5 E3404040   CC_MRTN_POINT

ENTER:  CURRENT DISPLAY
PF1 : UNDEFINED          PF2 : BROWSE TEMP STORAGE PF3 : UNDEFINED
PF4 : EIB DISPLAY        PF5 : WORKING STORAGE     PF6 : USER DISPLAY
PF7 : SCROLL BACK HALF   PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL   PF11: SCROLL FORWARD FULL PF12: REMEMBER DISPLAY
```

The most recent entry ends with the “ENDOF-TRACETBL” eye-catcher. The CICS Comet version number should follow immediately after the end of the trace table.

```
TRANSACTION: EMNU   PROGRAM: IVPCEMNU   TASK NUMBER: 0000359   DISPLAY:  00
ADDRESS: 00065530                    WORKING STORAGE+X'000540'
00065530   000000   C3D4C46D D3E5D36D E2C5E3E4 D7404040   CMD_LVL_SETUP
00065540   000010   D3D6C7C7 C9D5C740 40404040 40404040   LOGGING
00065550   000020   D3D6C75B C5E7C9E3 40404040 40404040   LOG$EXIT
00065560   000030   D3D6C75B 5BE7C9E3 40404040 40404040   LOG$$XIT
00065570   000040   C3C36DD4 C1C9D56D C2C5C6D6 D9C54040   CC_MAIN_BEFORE
00065580   000050   C3C36DD4 C1C9D56D D5D6C5E7 C9E3C240   CC_MAIN_NOEXITB
00065590   000060   C3C36DC6 C9D5C46D D4D6C4E4 D3C54040   CC_FIND_MODULE
000655A0   000070   C3C2D3C4 C6C8D4E2 40404040 40404040   CBLDFHMS
000655B0   000080   C3C2D36D E3C5E2E3 6DD9C5C7 F3404040   CBL_TEST_REG3
000655C0   000090   C3C2D36D C5E7C9E3 6DC5D9D9 D6D94040   CBL_EXIT_ERROR
000655D0   0000A0   C3C2D36D C5E7C9E3 6DD9F340 40404040   CBL_EXIT_R3
000655E0   0000B0   C3C2D3E7 C9E3D4E2 40404040 40404040   CBLXITMS
000655F0   0000C0   C3C35B5B C5C2D4E2 40404040 40404040   CC$$EBMS
00065600   0000D0   C3C36DC5 D4E26DE2 C5D5C4D4 C1D74040   CC_EMS_SENDMAP
00065610   0000E0   C5D5C4D6 C66OE3D9 C1C3C5E3 C1C2D3C5   ENDOF-TRACETABLE
00065620   0000F0   10C2D4E2 00000000 E5F2F4F1 00000000   .BMS....V244....

ENTER:  CURRENT DISPLAY
PF1  : UNDEFINED           PF2  : BROWSE TEMP STORAGE PF3  : UNDEFINED
PF4  : EIB DISPLAY         PF5  : WORKING STORAGE     PF6  : USER DISPLAY
PF7  : SCROLL BACK HALF    PF8  : SCROLL FORWARD HALF PF9  : UNDEFINED
PF10: SCROLL BACK FULL     PF11: SCROLL FORWARD FULL PF12: REMEMBER DISPLAY
```

**Locating the CICS Comet TCA Area**

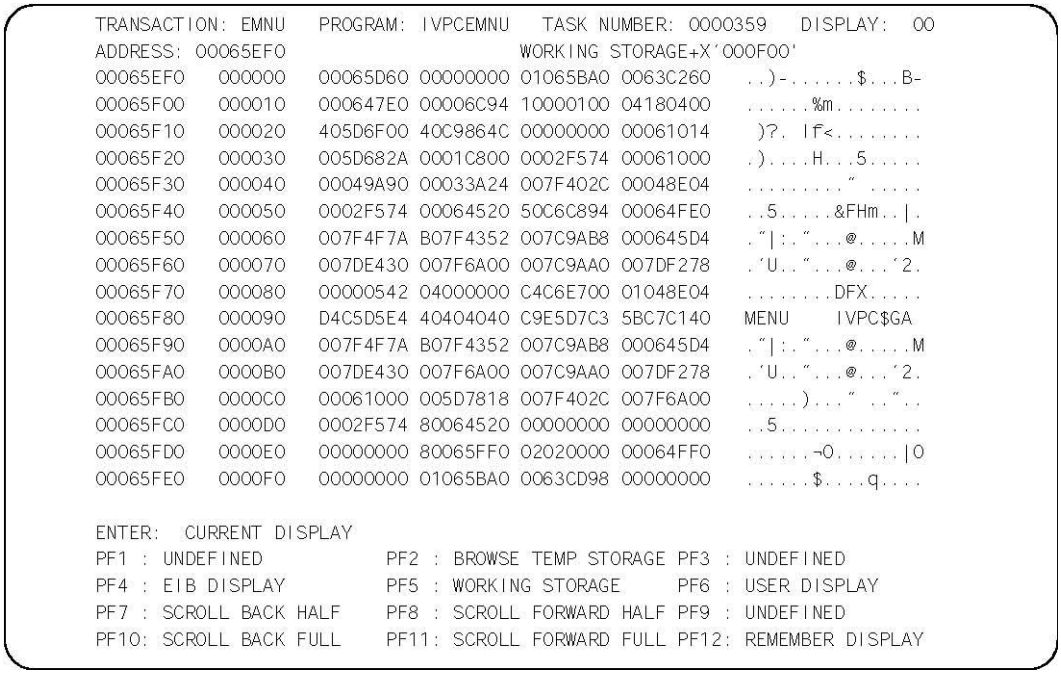
1. Follow the procedure outlined “CICS Comet’s EIS Trace Table” on page 106, to find the CICS Comet EIS area
2. Then take the address immediately following the CC$$EISx eye-catcher as the CICS Comet user TCA address. This address should be at offset x’0100' into the CICS Comet EIS storage area. The CICS Comet system TCA address will be at offset +0 into the CICS Comet EIS storage area. The CICS Comet system TCA address will be at offset +0 into the CICS Comet user TCA address.

In the screen below you can see that this is a CICS Comet EIS area by the CC$$EISM eye-catcher.

```
TRANSACTION: EMNU   PROGRAM: IVPCEMNU   TASK NUMBER: 0000359   DISPLAY:  00
ADDRESS: 00065060                    WORKING STORAGE+X'000070'
00065060   000000   000650DC 000650E2 00065F88 00000000   ..&...&S..¬h....
00065070   000010   00000000 00000000 00000000 00000000   ................
00065080   000020   00000000 00000000 00000000 00000000   ................
00065090   000030   00000000 80000000 00000000 00000000   ................
000650A0   000040   00000000 00000000 0C02E000 490000CC   ............ÈÈ..
000650B0   000050   00F0F1F0 F4F8C5F0 F4000000 00000000   .01048E04.......
000650C0   000060   00000000 00000000 00000000 00000000   ................
000650D0   000070   00000000 00000000 00065EF0 00000000   ..........;0....
000650E0   000080   00060000 0001C970 C3C35B5B C5C9E2D4   .....ICC$$EISM
000650F0   000090   00065EF0 04FEC5C9 D6D7E3E2 1804F000   ..;0..EIOPTS..0.
00065100   0000A0   29000100 00056204 000020F5 F0F5C4F7   ..........505D7
00065110   0000B0   C3C6F800 F5F0F5C4 F7C3C6F8 00000268   CF8.505D7CF8....
00065120   0000C0   C5C9C1D9 C7E20000 00065058 00000542   EIARGS....&.....
00065130   0000D0   04000000 C4C6E700 01048E04 D4C5D5E4   ....DFX.....MENU
00065140   0000E0   40404040 C9E5D7C3 5BC7C140 4CE3C3C1       IVPC$GA <TCA
00065150   0000F0   C3C3C3C1 007F4F7A B07F4352 007C9AB8   CCCA.“|:.“...@..

ENTER:  CURRENT DISPLAY
PF1  : UNDEFINED          PF2  : BROWSE TEMP STORAGE PF3  : UNDEFINED
PF4  : EIB DISPLAY        PF5  : WORKING STORAGE     PF6  : USER DISPLAY
PF7  : SCROLL BACK HALF   PF8  : SCROLL FORWARD HALF PF9  : UNDEFINED
PF10: SCROLL BACK FULL    PF11: SCROLL FORWARD FULL PF12: REMEMBER DISPLAY
```

The eye-catcher is followed by the CICS Comet TCA address, which is underlined in the example screen shown above.

In the screen below you can see the CICS Comet TCA area. The TCA address was underlined in the screen on the previous page. The TCA address of 00065EF0 was entered in the CEDF screen following the keyword “ADDRESS”.



```
TRANSACTION: EMNU   PROGRAM: IVPCEMNU   TASK NUMBER: 0000359   DISPLAY:  00
ADDRESS: 00065EF0                      WORKING STORAGE+X'000F00'
00065EF0   000000   00065D60 00000000 01065BA0 0063C260   ..)-......$...B-
00065F00   000010   000647E0 00006C94 10000100 04180400   ......%m........
00065F10   000020   405D6F00 40C9864C 00000000 00061014    )?. If<........
00065F20   000030   005D682A 0001C800 0002F574 00061000   .)....H...5.....
00065F30   000040   00049A90 00033A24 007F402C 00048E04   .........“ .....
00065F40   000050   0002F574 00064520 50C6C894 00064FE0   ..5.....&FHm..|.
00065F50   000060   007F4F7A B07F4352 007C9AB8 000645D4   .“|:.“...@.....M
00065F60   000070   007DE430 007F6A00 007C9AA0 007DF278   .'U..“...@...'2.
00065F70   000080   00000542 04000000 C4C6E700 01048E04   .........DFX.....
00065F80   000090   D4C5D5E4 40404040 C9E5D7C3 5BC7C140   MENU    IVPC$GA
00065F90   0000A0   007F4F7A B07F4352 007C9AB8 000645D4   .“|:.“...@.....M
00065FA0   0000B0   007DE430 007F6A00 007C9AA0 007DF278   .'U..“...@...'2.
00065FB0   0000C0   00061000 005D7818 007F402C 007F6A00   .....)...“ ..“..
00065FC0   0000D0   0002F574 80064520 00000000 00000000   ..5.............
00065FD0   0000E0   00000000 80065FF0 02020000 00064FF0   ......¬O......|O
00065FE0   0000F0   00000000 01065BA0 0063CD98 00000000   ......$....q....

ENTER:  CURRENT DISPLAY
PF1 : UNDEFINED           PF2 : BROWSE TEMP STORAGE PF3 : UNDEFINED
PF4 : EIB DISPLAY         PF5 : WORKING STORAGE     PF6 : USER DISPLAY
PF7 : SCROLL BACK HALF    PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL    PF11: SCROLL FORWARD FULL PF12: REMEMBER DISPLAY
```

**Finding the CICS Comet Release in the EIS Area**

The CICS Comet version number follows the trace table in the CICS Comet EIS storage area. The format of the version number is either V*nnn* or V*n*M*n*. For example: “V331” for version 5.4.0 or “V246” for version 2.4.6.

Enter the HALLEY VERSION command to display the CICS Comet version number. CICS Comet also detects any version number mismatches in the CICS Comet domains and displays them on the console during CICS Comet initialization.

```
TRANSACTION: EMNU   PROGRAM: IVPCEMNU   TASK NUMBER: 0000359   DISPLAY:  00
ADDRESS: 00065530                    WORKING STORAGE+X'00054O'
00065530   000000   C3D4C46D D3E5D36D E2C5E3E4 D7404040   CMD_LVL_SETUP
00065540   000010   D3D6C7C7 C9D5C740 40404040 40404040   LOGGING
00065550   000020   D3D6C75B C5E7C9E3 40404040 40404040   LOG$EXIT
00065560   000030   D3D6C75B 5BE7C9E3 40404040 40404040   LOG$$XIT
00065570   000040   C3C36DD4 C1C9D56D C2C5C6D6 D9C54040   CC_MAIN_BEFORE
00065580   000050   C3C36DD4 C1C9D56D D5D6C5E7 C9E3C240   CC_MAIN_NOEXITB
00065590   000060   C3C36DC6 C9D5C46D D4D6C4E4 D3C54040   CC_FIND_MODULE
000655A0   000070   C3C2D3C4 C6C8D4E2 40404040 40404040   CBLDFHMS
000655B0   000080   C3C2D36D E3C5E2E3 6DD9C5C7 F3404040   CBL_TEST_REG3
000655C0   000090   C3C2D36D C5E7C9E3 6DC5D9D9 D6D94040   CBL_EXIT_ERROR
000655D0   0000A0   C3C2D36D C5E7C9E3 6DD9F340 40404040   CBL_EXIT_R3
000655E0   0000B0   C3C2D3E7 C9E3D4E2 40404040 40404040   CBLXITMS
000655F0   0000C0   C3C35B5B C5C2D4E2 40404040 40404040   CC$$EBMS
00065600   0000D0   C3C36DC5 D4E26DE2 C5D5C4D4 C1D74040   CC_EMS_SENDMAP
00065610   0000E0   C5D5C4D6 C66OE3D9 C1C3C5E3 C1C2D3C5   ENDOF-TRACETABLE
00065620   0000F0   10C2D4E2 00000000 E5F2F4F1 00000000    .BMS....V331....

ENTER:  CURRENT DISPLAY
PF1 : UNDEFINED          PF2 : BROWSE TEMP STORAGE PF3 : UNDEFINED
PF4 : EIB DISPLAY        PF5 : WORKING STORAGE     PF6 : USER DISPLAY
PF7 : SCROLL BACK HALF   PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL   PF11: SCROLL FORWARD FULL PF12: REMEMBER DISPLAY
```

# Appendix C. Performance Monitoring

CICS Comet monitors internal translation activity. Enter the `HALLEY` command to display a summary of system-wide CICS Comet statistics. You must enter a space after `HALLEY`.

The following screen shows system wide statistics after the HALLEY command is entered. The line showing the number of programs that are Dynamically Attached refers to CICS Comet's ability to automatically add the CICS Comet command level stub to the beginning of application programs. CICS Comet's Dynamic Attach facility is controlled by the Include/Exclude members described in "Dynamic Attach for Macro-Level Translation" on page 17.

```
HALLEY
---------------------- CICS COMET SYSTEM WIDE STATISTICS ----------------------
================================================================================
TOTAL MACRO LEVEL REQUESTS CONVERTED BY CICS COMET.............: 88,540,007

FULL FUNCTION BMS MAPSETS MOVED TO XA ADDRESS SPACE BY COMET....: ____73,054
NUMBER OF DSA/SUBPOOL PAGES SAVED BY MOVING FULL FUNCTION MAPS.:  _____2,912
NUMBER OF DSA/SUBPOOL BYTES SAVED BY MOVING FULL FUNCTION MAPS.:  _1,243,756

FAST FUNCTION BMS MAPSETS MOVED TO XA ADDRESS SPACE BY COMET....: ____73,054
NUMBER OF DSA/SUBPOOL PAGES SAVED BY MOVING FAST FUNCTION MAPS.:  _____2,912
NUMBER OF DSA/SUBPOOL BYTES SAVED BY MOVING FAST FUNCTION MAPS.:  _1,243,756

NUMBER OF PROGRAMS ENCOUNTERED BY DYNAMIC ATTACH................: _______127
NUMBER OF DYNAMICALLY ATTACHED ASSEMBLER PROGRAM.................: ________20
NUMBER OF DYNAMICALLY ATTACHED PL/I PROGRAMS.....................: _________7
NUMBER OF DYNAMICALLY ATTACHED COBOL PROGRAMS....................: _______100
NUMBER OF DYNAMICALLY ATTACHED RESIDENT PROGRAMS.................: ________12
```

☞ BMS map statistics are available only on non-CICS/TS releases of Comet.

In addition to system-wide summary statistics, CICS Comet also provides a statistical detail screen that shows the number of requests by macro level type. Enter the following commands to display the statistical details screen:

HALLEY INQUIRY Detail statistics
HALLEY RESET Clear CICS Comet detail statistics

```
HALLEY INQUIRY
-------------- CICS COMET DETAIL STATISTICS ------------------
================================================================
         DLI REQUESTS CONVERTED BY CICS COMET:    _______23,546
         EIP REQUESTS CONVERTED BY CICS COMET:    ________4,368
         TCP REQUESTS CONVERTED BY CICS COMET:    ___________76
         FCP REQUESTS CONVERTED BY CICS COMET:    ____3,451,230
         TDP REQUESTS CONVERTED BY CICS COMET:    _______98,702
         TSP REQUESTS CONVERTED BY CICS COMET:    ________3,321
         SCP REQUESTS CONVERTED BY CICS COMET:    __________103
         PCP REQUESTS CONVERTED BY CICS COMET:    ________7,876
         ICP REQUESTS CONVERTED BY CICS COMET:    __________231
         KCP REQUESTS CONVERTED BY CICS COMET:    __________182
         JCP REQUESTS CONVERTED BY CICS COMET:    ______234,312
         SPP REQUESTS CONVERTED BY CICS COMET:    ________6,380
         BMS REQUESTS CONVERTED BY CICS COMET:    ____2,234,978
         TRP REQUESTS CONVERTED BY CICS COMET:    ________2,415
         DCP REQUESTS CONVERTED BY CICS COMET:    _______34,125
         DIP REQUESTS CONVERTED BY CICS COMET:    ________4,560
         BIF REQUESTS CONVERTED BY CICS COMET:    ___________34
```

# Appendix D. CICS Comet Program Stub

This appendix describes the characteristics of the CICS Comet stub attached to assembler, COBOL, and PL/I programs. The following table lists the location and length of the stub based upon the language the program was written that has an attached CICS Comet stub.

| Program Language | Stub Location | Stub Length (Hexadecimal) |
|---|---|---|
| Assembler | Beginning of load module | 30 bytes |
| VS/COBOL command level | Beginning of load module | 80 bytes |
| VS/COBOL macro level | Beginning of load module | 40 bytes |
| COBOL/LE | End of load module | 40 bytes |
| COBOL II | End of load module | 40 bytes |
| PL/I | End of load module | 30 bytes |

# Index


16 Mb Line .....38
Appendix A. About CICS Comet .....83
  Common Abend codes .....92
  Messages and Codes .....85
  Restrictions .....83
  Sample Programs .....96
Appendix B. Commands and Debugging .....98
  Comet Commands .....98
  Finding CICS Comet Release in EIS Area 116
  Finding Comet EIS Trace Table .....109
  Haly Service Transaction .....101
  Locating CICS Comet Trace Table .....112
  Locating CICS Comet TSA Area .....114
Appendix C. Performance Monitoring .....117
Appendix D. CICS Comet Program Stub .....119
CCCMDEXC Exclude Table .....42
CCCMDINC Include Table .....41
  CSA .....41
  CWA .....41
  TWA .....41
CCMACINC Include Table .....19
  FIX CBL .....20
  FIX CSA .....20
  FIX RMI .....20
  Mixed mode .....19
  RMODE .....20
CCSIPARM .....30
  Parameters .....32
CICS Comet Exits .....79
  How to use .....82
CMACEXC Exclude Table .....21
  CCMACEXC Entries .....22
Comet SAMPLIB Members
  Comet Samplib members .....94
Command Level
  CSA Address .....40
Customization .....27
Debugging .....43
  Abending PSW and Registers .....49
  ASRA Fact sheet .....52
  CEDF with CICS/COMET .....45
  Decoding Storage Operands .....53
  Program debugging with Comet domain .....54
  Summary .....48
Environment Built to Support Macro Level .....55
EY REFRESH Macro Programs (CC$$NEWC)
  HALLEY REFRESH .....102
Finding the Program Offset
  Finding program offset .....47
Initializing Dynamic Attach .....16
Installation ..... 7
  CC$$V540 .....16
  Checklist ..... 7
  DFHRPL .....15
  License Key .....37
  New Usner .....10
  PLT Entries .....12
  PPT and PCT entries .....10
  Startup library placement .....13
  Step 1 Unload the First File ..... 8
  Step 2 Copy distribution libs ..... 9
  Step 3 Install Comet .....10
  Step 4 Startup Overrides .....14
  Step 5 Update CICS startup .....15
  Step 6 Initialize Dynamic Attach .....16
  Step 7 Dynamic Attach for Macro lvel .....17
  Step 8 CCMACEXC Table .....23
  Step 9 Starting CICS Comet .....24
  Step 10 IVP .....25
  Upgrade .....11
Installation Interface .....63
  Cobol II and database .....67
  Coding Non-XA DCT .....72
  Coding Non-XA FCT .....71
  Coding Non-XA JCT .....73
  Coding Non-XA PCT .....74
  Coding Non-XA PPT .....75
  Coding Non-XA TST .....77
  Coding Remote DCT .....68
  Coding remote FCT .....68
  Direct CICS Control block search support...71
  Function Shipping .....68
  IDMS .....63
  ISAM .....78
  System 2000 DB .....66
  TOTAL/DB .....65
Installation Non-XA TCT .....76
Making Dynamic Changes to Tables .....17
Moving BMS above the line .....38
MSA .....29
Notis .....29
PL/I .....28
Relink for BMS .....38
Selecting Programs to Attach .....17
  Displaying included and excluded programs
    .....17

Matching your programs....17
Special Procedures....57
Relink JCL Sample....60
Relinking Macro Level Programs....58
Support Macro
Comet stub and program offsets....56
CSA/CWA....55
FCT....56
PCT....56
PPT....56
TCA/TWA....55
TCTTE....56
Top Secret....13
Virtual Storage Constraint Relief .... 2
XA/RELO....13