Compuware

THE TECHNOLOGY
PERFORMANCE COMPANY

Xpediter/CICS

PL/I User's Guide

Release 9.1

Please direct questions about Xpediter/CICS
or comments on this document to:

Xpediter/CICS Customer Support

http://frontline.compuware.com/

Doc. CWXCUP9B

June 21, 2011

Contents

Figures

Introduction

This guide provides step-by-step instructions on how to use Xpediter/CICS to solve common debugging problems. To become acquainted with Xpediter/CICS, you should first review Chapter 1, “Product Overview” and Chapter 2, “Getting Started”.

The chapters listed below contain exercises using common examples. The first examples show how to access and exit the product and how to debug a simple transaction. Subsequent chapters provide exercises on how to set breakpoints in a program, how to step through the program, and how to change storage. The later chapters provide exercises on advanced features.

The exercises cover only a portion of the many Xpediter/CICS screens and functions. For more information on these topics, see the *Xpediter/CICS Reference Manual*.

This guide contains the following chapters:

Chapter 1, “Product Overview”: Introduces Xpediter’s facilities and discusses the need for an interactive debugging tool. The overview also includes features that are new to this release of the product.

Chapter 2, “Getting Started”: Shows how to prepare your application for debugging, accessing, and exiting Xpediter/CICS.

Chapter 3, “Testing a PL/I Program”: Shows how to test a PL/I program with and without breakpoints, and how to inspect program data.

Chapter 4, “Debugging Applications Without Source Code”: Provides tips on performing sourceless debugging.

Chapter 5, “Debugging Subroutines”: Provides tips on setting breakpoints in the calling and called programs and selecting and excluding CSECTs.

Chapter 6, “Analyzing Program Execution”: Describes how to set up an analysis of a program.

Chapter 7, “Monitoring Tasks Started from Remote Terminals”: Describes how to set and intercept remote traps.

Chapter 8, “Providing Storage Protection”: Shows how to handle storage violations and set region-wide storage protection.

Chapter 9, “Interfacing with Abend-AID for CICS”: Discusses the interface with Compuware’s Abend-AID for CICS fault diagnosis product.

Chapter 10, “Using Automatic Trap Activation”: Explain how ATA traps abends without the user having an Xpediter session active.

Chapter 11, “Setting Up a Profile”: Describes how to set up a customized environment that modifies program default values.

Chapter 12, “Accessing Files”: Shows you how to browse records, change file service requests, and work with data and storage queues.

Chapter 13, “Accessing DL/1 Databases”: Shows how to list PCBs and edit a DL/1 segment.

Chapter 14, “Using Xpediter/CICS with DB2”: Tips on row and column editing of DB2 tables and views.

Chapter 15, "Using Xpediter/CICS with MQ": Describes the MQ File Utility and how to debug MQ programs.

Chapter 16, "Accessing CICS Storage": — Describes how to access and update CICS storage.

Chapter 17, "Using Global Storage Protection": Shows how to set protection, define system labels, and control storage exceptions.

Chapter 18, "Editing CICS Tables and Control Blocks": Describes how to display CICS table entries and DSECTs.

Chapter 19, "Using 3270 Web Bridge Support": Demonstrates how to run Xpediter using the 3270 Web Bridge function.

"Glossary": Describes Xpediter/CICS features, and defines key terms.

Intended Audience

Xpediter/CICS is structured in three levels to accommodate several types of users. The audience for this manual may differ, depending on the experience of the user and the transactions for which the user is authorized. The following three transactions are used to describe Xpediter's three levels:

- **XPED** – Intended for application programmers who want a source level focus.
- **XPRT** – Intended for application programmers who want a break/abend focus.
- **XPSP** – Intended for experienced system programmers who are authorized to update CICS tables and control areas.

System Environment

System Requirements

Use of Xpediter/CICS requires the following:

- z/OS Release 1.6 or above operating environment
- CICS Transaction Server Release 3.1, 3.2, 4.1, or 4.2
- Compuware Shared Services Release 8.7 or above
- License Management System:
  - Minimum: Release 3.1 with PTFs
  - Recommended: Release 4.0 or above

Supported Environments

Note: This information is current as of the publication date. For the most recent information, go to Frontline. You will find the latest Release Notes listed with the other product documentation.

Xpediter/CICS supports the following:

- Dynamic Transaction Routing with the CICSPlex SM component of CICS Transaction Server
- The following programming languages (under runtime Language Environment [LE] for OS/390 2.10 and above in compatibility mode):

- Enterprise COBOL Releases 3.1, 3.2, 3.3, 3.4, 4.1, and 4.2
- COBOL for OS/390 & VM Releases 2.1 and 2.2
- COBOL for MVS & VM Release 1.2
- Enterprise PL/I Releases 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, and 4.1
- PL/I for MVS & VM Release 1.1
- z/OS C Releases 1.2 through 1.6
- z/OS XL C Releases 1.7 through 1.12
- OS/390 C Release 2.10

Note: For COBOL, compatibility means that Xpediter/CICS supports COBOL II and COBOL/370 programs that have been recompiled with COBOL for MVS & VM, COBOL for OS/390 & VM, or Enterprise COBOL. For PL/I, compatibility means that Xpediter/CICS supports OS PL/I programs that have been recompiled with PL/I for MVS & VM or Enterprise PL/I. For C, compatibility means that Xpediter/CICS supports programs that have been recompiled using the supported C compilers listed above.

- The following non-LE programming languages:
  - High Level Assembler
  - Assembler H Version 2
  - COBOL/370 Release 1.1
  - VS COBOL II Releases 1.3.1, 1.3.2, and 1.4
- The following IMS and DB2 releases:
  - IMS Releases 7.1, 8.1, 9.1, 10.1, and 11.1
  - DB2 Releases 7.1, 8.1, 9.1, and 10.1
- The following WebSphere MQ (formerly MQSeries) releases:
  - MQSeries for OS/390 Releases 2.1 and 5.2
  - WebSphere MQ for z/OS Releases 5.3, 6.0, and 7.0

Xpediter/CICS requires no modifications to CICS control programs. Installation requires updating CICS resources in the same manner as any CICS application package.

Xpediter/CICS cannot be used to debug executable modules that reside in CA Technologies CA-Panexec® libraries.

Related Publications

The documents in the following list are provided on CD-ROM with the Xpediter/CICS system. For details, see “Online Documentation” below. The *Installation Guide* and *Quick Reference* are also provided on paper.

- *Xpediter/CICS Installation Guide*: Gives step-by-step instructions for the system programmer to install, customize, and maintain Xpediter/CICS.
- *Xpediter/CICS Reference Manual*: Provides specific reference information about Xpediter’s features, utilities, menus, and command parameters.
- *Xpediter/CICS User’s Guide for Assembler, COBOL, PL/I, or C*: Introduction to Xpediter’s levels, screens, and functions.
- *Xpediter/CICS Quick Reference*: Handy reference for screen IDs and command syntax.
- *Xpediter/CICS Messages and Codes*: Lists error and warning messages that might be encountered during installation or use of Xpediter.

For information regarding Compuware Shared Services, refer to the *Enterprise Common Components Installation and Customization Guide*.

FrontLine Support Website

Access online customer support for Compuware products via our FrontLine support website. View or download documentation, frequently asked questions, and product fixes, or directly e-mail Compuware with questions or comments. To access FrontLine, you must first register and obtain a password at **http://frontline.compuware.com**.

Online Documentation

The installation package includes the following Xpediter documentation in these electronic formats:

- Release Notes in HTML format
- The product manuals in PDF format
- The Adobe PDF index file (PDX file).

View and print the PDF files with Adobe Reader. Download a free copy of the reader from Adobe's Web site, **http://www.adobe.com**.

The Xpediter product manuals are also available in HTML and IBM BookManager formats at the FrontLine Web site.

HTML files can be viewed with any standard web browser.

BookManager softcopy files can be viewed with any version of IBM BookManager READ or the IBM Library Reader. To learn more about BookManager or download the free Library Reader, go to **http://www.ibm.com**.

World Wide Web

Compuware's site on the World Wide Web provides information about Compuware and its products. The address is **http://www.compuware.com**.

Customer Support

Compuware provides a variety of support resources to make it easy for you to find the information you need.

FrontLine Support Web Site

You can access online information for Compuware products via our FrontLine support site at **http://frontline.compuware.com**.

FrontLine provides access to critical information about your Compuware products. You can review frequently asked questions, read or download documentation, access product fixes, or e-mail your questions or comments. The first time you access FrontLine, you are required to register and obtain a password. Registration is free.

Compuware now offers User Communities, online forums to collaborate, network, and exchange best practices with other Compuware solution users worldwide. To join, go to **http://groups.compuware.com**.

Contacting Customer Support

Phone

- USA and Canada: 1-800-538-7822 or 1-313-227-5444.
- All other countries: Contact your local Compuware office. Contact information is available at **http://frontline.compuware.com**.

Web

You can report issues via the **Report and Track Calls** tab on the FrontLine home page.

Note: Please report all high-priority issues by phone.

Mail

Xpediter/CICS Customer Support
Compuware Corporation
One Campus Martius
Detroit, MI 48226-5099

Corporate Web Site

To access Compuware's site on the Web, go to **http://www.compuware.com**.

The Compuware site provides a variety of product and support information.

Information for Customer Support

If problems arise, please check your manual for assistance. If problems persist, please obtain the following information before calling Compuware for assistance. This information will help determine the exact cause of the problem as quickly as possible.

1. Identify the release number of Compuware product(s) in use.
2. Identify the operating system.
3. Identify the release of CICS Transaction Server that is being used.
4. If an abend occurs, note the displacement and the module in which it occurs. If possible, obtain a copy of the system dump.
5. Note the sequence of steps (including all commands issued) that resulted in the problem. Also note any variable data types and programming languages involved.
6. To receive product fixes electronically, be ready to provide your email address.

Chapter 1. Product Overview

Xpediter/CICS gives the CICS programmer complete control over the execution of application code, trapping of abends, and access to data files and CICS storage, including tables and control blocks in DSECT format. An easy-to-use architecture allows you to interactively debug application programs quickly and accurately.

Xpediter/CICS lets you control the execution of your program and monitor its status at any time. You can set breakpoints to suspend execution (with or without conditions), change the program logic flow, intercept abends or storage violations, and many other functions, all without leaving the test session or recompiling.

Xpediter/CICS allows you to interact directly with the program as it executes, allowing you to perform the following functions:

- View and interact with program source code, online.
- Display and update records in files, transient data, temporary storage, DB2 tables, and DL/1 databases. Xpediter also gives you the option of logging any changes.
- List, browse, and update MQ message queues.
- List a task's active channels and containers, and browse container data.
- Stop execution of a program at any point and examine working storage.
- Resume execution at any point in the program.
- Execute statements one at a time while examining program logic.
- Modify any unprotected program data, CICS table, or CICS area.
- Monitor remote transactions.
- Ensure region integrity through storage protection.

Note: XPLINK programs are excluded from Xpediter/CICS storage protection.

Using Xpediter/CICS, you can observe a program as it executes, stop execution, look at intermediate results, correct problems as they arise, and proceed with the test. When Xpediter/CICS traps an abend, it displays a wealth of information that aids in understanding and correcting that abend. You can even resolve multiple problems during a single test session. Xpediter's Script Facility allows you to record selected primary and line commands entered during a debugging session, save them in a dataset, and then replay them later.

Xpediter/CICS provides alternatives for problem resolution. When the product identifies a problem, you can select particular statements at which to suspend execution, then analyze both working storage and a program trace. If a program stops at an abend, you can correct the problem or bypass the abend. To bypass an abend, just instruct Xpediter to resume execution at a statement number or offset past the point of the abend.

Three restricted modes of operation — Diagnosis Mode, Utilities Mode, and Diagnosis/Utilities Mode — allow a site to tailor its Xpediter implementation to suit the processing integrity and throughput requirements of its various CICS regions. This means you can deploy Xpediter/CICS as a crucial, safe, and effective tool in your production CICS regions. Utilizing Xpediter in a restricted operating mode, especially along with Abend-AID for CICS, lets you quickly diagnose and resolve critical production failures without wasting time and effort recreating the problem in a test region.

Note: All descriptions in this manual, unless otherwise noted, pertain to Xpediter's standard, non-restricted operating mode.

Product Architecture

Xpediter/CICS is designed to provide different levels of functionality.

The ISPF-like nature of Xpediter and its source-based approach make it immediately familiar. Yet it is powerful enough to meet your most complex debugging requirements. Xpediter/CICS provides an efficient, non-intrusive architecture that can protect the CICS region from storage violations. It also offers complete CICS-specific debugging capabilities for both application and system programmers in the CICS environment.

Xpediter/CICS supports applications written in Assembler, COBOL, C, or PL/I, plus a variety of other programming languages on an Assembler instruction-level basis.

Xpediter's unique multi-access design lets users access the product through their choice of three separate and securable debugging transactions:

- **XPED** access causes Xpediter to display the Source Listing screen (2.L) when an abend occurs or a breakpoint is encountered.
- **XPRT** retains the feel of earlier releases by displaying the Break/Abend screen (2.1) at an abend or breakpoint.
- **XPSP** access provides all the power of the XPED and XPRT transactions, plus additional system maintenance capabilities for the CICS specialist responsible for supporting the CICS region.

XPED is the standard transaction used by application programmers. All the commands and functions available with XPED can also be accessed through the XPRT transaction. With either transaction, Xpediter/CICS provides:

- Display-only access to CICS storage areas
- Update capability for application transaction storage areas
- Access to FCT or CEDA-defined files, transient data, temporary storage, IMS databases, DB2 tables, and MQ message queues.

The only real difference between the two transactions is that when a breakpoint is reached or an abend occurs, by default XPED displays the Source Listing screen (2.L), while XPRT displays the Break/Abend screen (2.1).

The XPSP transaction is designed for those system programmers authorized to update unprotected CICS tables and control areas. It permits unrestricted storage area updates and provides hung task analysis, along with other special region maintenance and debugging functions. XPSP allows you to establish and control system-wide storage protection.

Xpediter can also be configured to operate in any of three restricted modes of operation:

- Diagnosis Mode
- Utilities Mode
- Diagnosis/Utilities Mode.

These modes allow a site to tailor its Xpediter implementation to suit the processing integrity and throughput requirements of its various CICS regions. In Diagnosis Mode, the user is prevented from modifying data or changing the sequence of program execution. In Utilities Mode, only the Xpediter file utility, storage display facility, and source listing utility are accessible. Storage, databases, and files can be viewed and modified, but trap, trace, and monitor functions are unavailable. Diagnosis/Utilities Mode combines the restrictions of the other two modes. You can access Xpediter's file utility and source listing utility and view storage on the Memory Display (2.2 and 9.2), Task Storage Display (2.S), and CICS DSECTs (2.D and 9.D) screens. A mode indicator message is displayed in the upper left-hand corner of all Xpediter screens when the product is operating in one of the restricted modes. For more information Chapter 5, “Restricted Operating Modes” in the *Xpediter/CICS Reference Manual*.

Note: All descriptions in this manual, unless otherwise noted, pertain to Xpediter’s standard, non-restricted operating mode.

XPED and XPRT Transactions

The XPED transaction accesses all of Xpediter/CICS’s functions for source-level testing and debugging of your application program. The XPRT transaction is intended for CICS application programmers who prefer the break/abend approach to testing and debugging that was used in earlier releases of Xpediter/CICS. Both transactions provide the same enhanced debugging features and allow you to browse CICS storage areas. XPED and XPRT also let you specify storage protection options, abend traps, and trace options. You can customize your debugging session through profile options that you can store and use again. If an abend occurs, Xpediter/CICS automatically displays by default either the Source Listing screen (2.L) for XPED users, or the Break/Abend screen (2.1) for XPRT users.

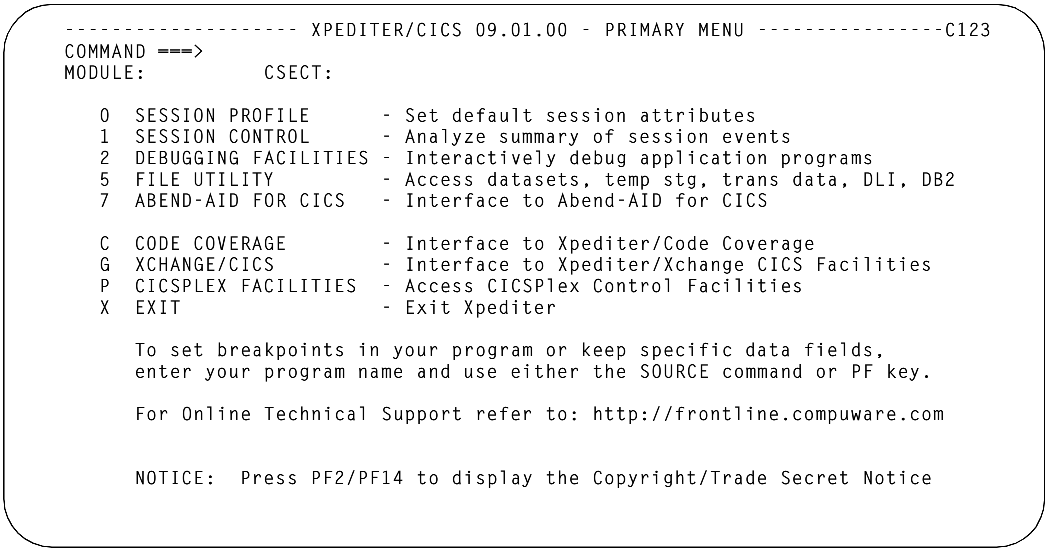
The Xpediter/CICS Primary Menu, shown in Figure 1-1, lists the screen number and name of each product facility menu available via the XPED and XPRT transactions. The option shown for Code Coverage, a sophisticated and powerful tool used with Xpediter/CICS to verify the thoroughness of testing, is only displayed if the separate Xpediter/Code Coverage product has been installed for the current CICS region. The option shown for Xchange/CICS is only displayed if Xpediter/*Xchange* is active in the CICS region. The option shown for CICSPLEX facilities is only displayed if the CICSPLX global parameter is set to YES.

There are four main ways you can navigate through Xpediter/CICS:

- Type one of the screen numbers from a displayed menu in the COMMAND field and press Enter.
- Move the cursor next to the desired screen number on a menu, type **S**, and press Enter.
- Transfer directly to any product screen by typing an equal sign (=) followed by the complete screen number — for instance **=5.2.2** — then pressing Enter.
- Use PF keys to transfer to commonly used screens. The PF keys and defaults for the corresponding screens are as follows:

PF6	LOCATE * (2.L or 2.20)
PF13	Primary Menu
PF14	Memory Display screen (2.2)
PF16	Variable Storage screen (2.3)
PF17	Program Trace screen (2.4)
PF21	File Utility Menu (5).

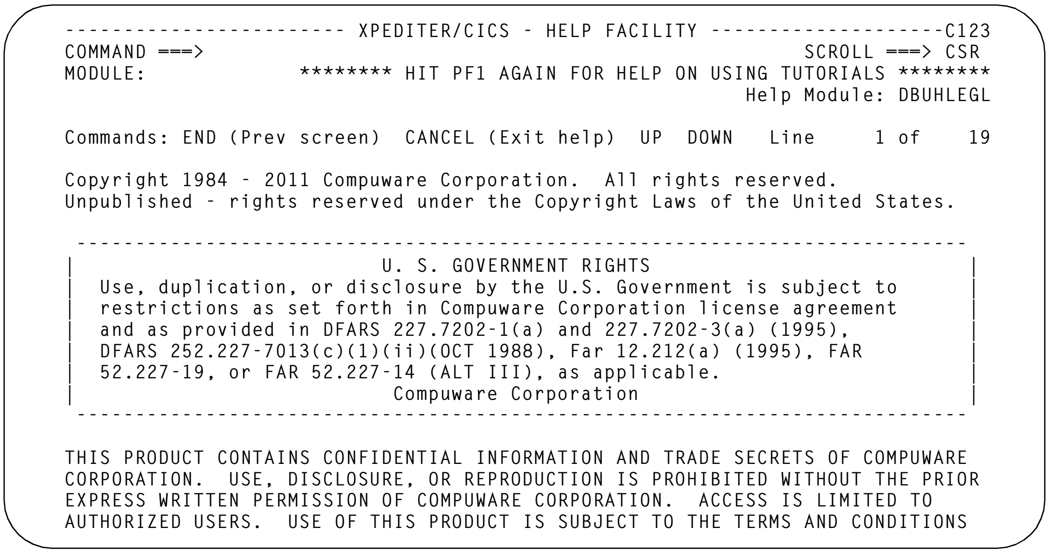
Figure 1-1. Primary Menu (XPED/XPRT)



```
-------------------- XPEDITER/CICS 09.01.00 - PRIMARY MENU ----------------C123
COMMAND ===>
MODULE:          CSECT:

   0  SESSION PROFILE      - Set default session attributes
   1  SESSION CONTROL      - Analyze summary of session events
   2  DEBUGGING FACILITIES - Interactively debug application programs
   5  FILE UTILITY         - Access datasets, temp stg, trans data, DLI, DB2
   7  ABEND-AID FOR CICS   - Interface to Abend-AID for CICS

   C  CODE COVERAGE        - Interface to Xpediter/Code Coverage
   G  XCHANGE/CICS         - Interface to Xpediter/Xchange CICS Facilities
   P  CICSPLEX FACILITIES  - Access CICSPlex Control Facilities
   X  EXIT                 - Exit Xpediter

      To set breakpoints in your program or keep specific data fields,
      enter your program name and use either the SOURCE command or PF key.

      For Online Technical Support refer to: http://frontline.compuware.com


      NOTICE:  Press PF2/PF14 to display the Copyright/Trade Secret Notice
```

While the Primary Menu is displayed, pressing PF2 or PF14 will display the copyright and trade secret notices as shown in Figure 1-2.

Figure 1-2. Copyright/Trade Secret Notice Screen



```
------------------------ XPEDITER/CICS - HELP FACILITY --------------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE:             ******** HIT PF1 AGAIN FOR HELP ON USING TUTORIALS ********
                                                               Help Module: DBUHLEGL

Commands: END (Prev screen)  CANCEL (Exit help)  UP  DOWN   Line     1 of    19

Copyright 1984 - 2011 Compuware Corporation.  All rights reserved.
Unpublished - rights reserved under the Copyright Laws of the United States.

 ----------------------------------------------------------------------------
|                          U. S. GOVERNMENT RIGHTS                           |
|  Use, duplication, or disclosure by the U.S. Government is subject to      |
|  restrictions as set forth in Compuware Corporation license agreement      |
|  and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995),          |
|  DFARS 252.227-7013(c)(1)(ii)(OCT 1988), Far 12.212(a) (1995), FAR         |
|  52.227-19, or FAR 52.227-14 (ALT III), as applicable.                     |
|                           Compuware Corporation                            |
 ----------------------------------------------------------------------------

THIS PRODUCT CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF COMPUWARE
CORPORATION.  USE, DISCLOSURE, OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR
EXPRESS WRITTEN PERMISSION OF COMPUWARE CORPORATION.  ACCESS IS LIMITED TO
AUTHORIZED USERS.  USE OF THIS PRODUCT IS SUBJECT TO THE TERMS AND CONDITIONS
```

XPSP Transaction

The XPSP transaction gives the CICS system programmer all the functionality of the XPED and XPRT transactions, and adds special CICS region-related utilities. When you access Xpediter/CICS with XPSP, the Primary Menu displayed is similar to the XPED/XPRT Primary Menu, but with one additional option:

```
9  SYSTEM FACILITIES    - Access global region analysis facilities
```

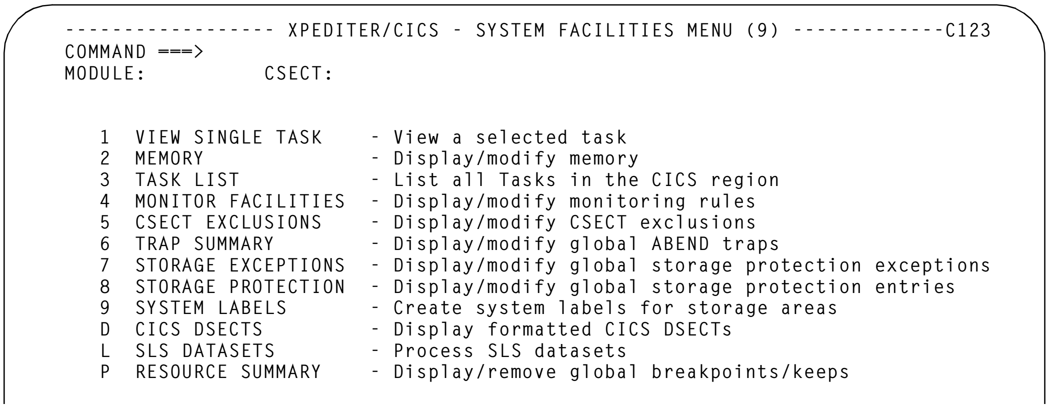
When you select option 9, System Facilities, you have access to the extended XPSP functions, which include:

- A CICS storage and table editor

- Power to activate storage protection anywhere in the CICS region
- Ability to analyze hung (system suspended) CICS transactions and chain through CICS storage areas
- Ability to open and close source listing datasets.

The System Facilities Menu (Figure 1-3) shows the additional functions available to the XPSP user.

Figure 1-3. System Facilities Menu (9) for XPSP Users



```
----------------- XPEDITER/CICS - SYSTEM FACILITIES MENU (9) ------------C123
COMMAND ===>
MODULE:          CSECT:


   1  VIEW SINGLE TASK    - View a selected task
   2  MEMORY              - Display/modify memory
   3  TASK LIST           - List all Tasks in the CICS region
   4  MONITOR FACILITIES  - Display/modify monitoring rules
   5  CSECT EXCLUSIONS    - Display/modify CSECT exclusions
   6  TRAP SUMMARY        - Display/modify global ABEND traps
   7  STORAGE EXCEPTIONS  - Display/modify global storage protection exceptions
   8  STORAGE PROTECTION  - Display/modify global storage protection entries
   9  SYSTEM LABELS       - Create system labels for storage areas
   D  CICS DSECTS         - Display formatted CICS DSECTs
   L  SLS DATASETS        - Process SLS datasets
   P  RESOURCE SUMMARY    - Display/remove global breakpoints/keeps
```

Product Facilities

Compuware product developers know that, without Xpediter, debugging an application can be among the most time-consuming and exasperating tasks you perform. Yet an error-free application is an absolute necessity. The Xpediter/CICS facilities listed below help you locate, identify, and eliminate application errors.

Source-Level Testing and Debugging Facilities

- Include an interactive, source code display for PL/I, COBOL, C, and Assembler programs
- Allow dynamic interaction with program source listings
- Step through (execute) program statements and instructions in timed slow motion or one at a time (single-stepping)
- Stop execution before or after specified statements or instructions
- Skip or redirect execution around specified statements or instructions
- Detect, isolate, and prevent transaction abends and storage violations
- Allow interactive changes to program variables

Note: In its character display of memory, Xpediter/CICS uses a decimal point to stand for any unprintable character. For that reason, a decimal point you type into that character display will not be recognized unless it overtypes a printable character — ***not another decimal point***. When hex is displayed, **x'4B'** can be entered to denote a decimal point.

- Bypass or modify faulty logic flow
- Provide a statement-level trace
- Count executions of repetitive statements or paragraphs
- Record, save, and play back scripts of primary and line commands.

- Interface dynamically with Compuware's fault diagnostic product Abend-AID for CICS.
- View CICS resources online.

CICS Storage Protection Facilities

- Region-wide protection capability
- Protection filters by terminal, transaction, and program
- Protection exceptions plus interactive and unattended protection options
- Over 90 detailed diagnostics from Xpediter/CICS
- Full range of Abend-AID for CICS diagnostics.

CICS Region Maintenance Facilities

- CICS storage and table editor
- Formatted CICS DSECTs
- Region-wide trap and trace
- Hung transaction analysis
- Storage chains.

File Utility

- Browse, edit, and map records (for COBOL and PL/I only) from CICS files
- Log changes to supported resources
- Support for BDAM and VSAM files, temporary storage, transient data, DB2 and IMS databases, and MQ queues.

Customizing Facilities

- Multitransaction architecture with specific facilities for different types of users
- Three restricted modes of operation
- User session profiles
- Hexadecimal calculator
- Screen footings.

Xpediter/CICS can be used in both the test and production environments for a variety of tasks, including:

- **Storage Protection** — Protection can be set up to monitor new transactions or programs in the production region, and intercept storage violations before they occur.

  Note: XPLINK programs are excluded from Xpediter/CICS storage protection.

- **Bad Record Correction** — The File Utility can be used in either environment to correct bad records.
- **System Maintenance** — Authorized users can view and modify CICS storage without bringing down the region.
- **Hung Task Analysis** — System programmers can look at an end user's task to determine problems.
- **Remote Trapping** — Application support personnel are able to set traps to gain control and diagnose end-user sessions.

Help Facility

Xpediter/CICS contains an extensive set of Help screens to assist you in learning the product. The Xpediter/CICS Help facility uses hypertext links to allow easy navigation through topics. Links to related topics are displayed with distinctive color and intensity. Simply tab to the hypertext link and press Enter to display the linked topic.

Help screens are available from any product screen by pressing PF1. The screens provide detailed information about the commands and functions available in each of the Xpediter/CICS areas. Use the Help screens when you need information about a particular screen or command. For example, to obtain a description of the SHOW command, type SHOW on the COMMAND line and press PF1, or enter the primary command HELP SHOW. A description of the format and use of the SHOW command will be displayed.

While in the Help facility, use the following keys to move through the screens:

PF3	Return to the previous Help topic
PF7	Scroll backward
PF8	Scroll forward

Many common 3270 terminal emulation software packages running on PCs can be configured to allow selection of hypertext links by double-clicking with the left mouse button. You can also double-click on the highlighted commands (CANCEL, END, UP, and DOWN) at the top of the Help screens to return to previous topics, scroll, or exit the Help facility.

Users of IBM Personal Communications/3270 Version 4 or above should perform the following steps to enable mouse navigation of the Xpediter/CICS Help facility:

1. On the Assist menu, click Hotspots Setup.
2. Click the Point-and-Select (Enter at cursor position) check box and then click OK.

The Help PF key can also be used to supply you with additional error information. Error messages are usually enclosed in asterisks (*). When an error message is displayed with plus signs (+), you can receive more specific information about that message by pressing PF1. For example, pressing PF1 for the error message **NO SOURCE AVAILABLE** will provide exact information on why source for that particular program cannot be located.

HELP NEWS provides online information about the new features in the latest release, including new commands, screens, and parameters. HELP NOSOURCE provides information to help you discover why source listings are not displayed for a particular program. Several possible causes are listed with suggested solutions.

Product Conventions

This section discusses the following Xpediter/CICS product conventions:

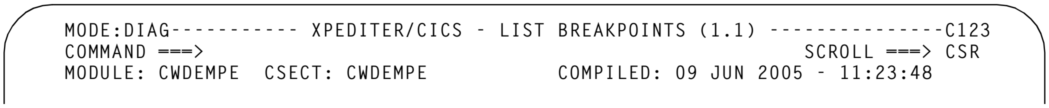
- Common screen fields
- Command entry
- PF key settings
- Update password security.

Common Screen Fields

Each screen is identified on the top line by a title or a screen ID. You can navigate from one Xpediter/CICS screen to another using menus, screen IDs (fast path), or commands.

Figure 1-4 shows the fields that are common across Xpediter/CICS screens. Each field is described below.

Figure 1-4. Common Screen Fields



```
MODE:DIAG---------- XPEDITER/CICS - LIST BREAKPOINTS (1.1) --------------C123
COMMAND ===>                                                SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48
```

- **Mode Indicator Message** — If Xpediter is operating in one of it's restricted modes, a message is displayed in the upper left corner of all screens.
- **Screen Title and ID** — Each screen title and ID is unique. Specify the screen ID in the COMMAND field to display that screen. For example, **=1.1** displays the List Breakpoints screen. A portion of the title is overlaid with the current date and time if the SET SUPPORT ON command has been issued.
- **System ID** — The SYSIDNT value for the current CICS region is conveniently displayed in the upper-right corner of every screen.
- **COMMAND Field** — Type primary commands and screen IDs in the COMMAND field. Some PF keys also depend on the parameters that you enter in this field.

  Multiple primary commands, separated by the current command delimiter, can be entered at the same time. A semi-colon (;) is the default delimiter.

  Note: The default command delimiter can be changed with the CMDDLM global parameter. (See the Xpediter/CICS Installation Guide chapter entitled "Global Parameters Table" for more information.) The CMDDLM value is the default delimiter only for users with no delimiter stored in their user profile, such as new users or existing users invoking a new release of Xpediter for the first time. Once a command delimiter has been set for a given user, it can be changed using the DELIM field on the Set Profile Defaults screen (0.1). See the *Xpediter/CICS Reference Manual* for details.

  Note: The HELP primary command cannot be entered together with other commands, and multiple primary commands cannot be entered from Xpediter's Help facility.

- **SCROLL Field** — Sets the scroll value for screens that permit scrolling.
- **MODULE Field** — Displays the current load module name regardless of what CSECT within the load module is being debugged.
- **CSECT Field** — Identifies the CSECT on the Xpediter screen. To change to another CSECT, simply type the new name. Current breakpoints and keeps are retained until they are deleted or the session is ended.

Notes:

- The CSECT field accomodates mixed-case CSECT names up to 16 characters and abbreviated CSECT names when the name is greater than 16 characters.
- This field is protected on the Script Dataset Allocation screen (0.6), the Data Area screens (2.3), and the Break/Abend screen (2.1).
- Xpediter/CICS may issue diagnostic or informational messages that overlay the CSECT and COMPILED fields. If the message is surrounded by plus signs (+++), you can access extended help by pressing PF1. When you press PF3 to exit the help, the CSECT and COMPILED fields are redisplayed. If extended help is not available, simply press Enter to redisplay the fields.

- **COMPILED: Field** — Displays the date and time the program was compiled in DD MMM YYYY format. When Xpediter/CICS issues a diagnostic or informational message, the message displays instead of the compilation date and time.

When the text of the message is surrounded by plus signs (++), type HELP in the COMMAND field and press Enter, or use PF1 (HELP), to obtain more detailed information about the message.

Command Entry

Xpediter/CICS has many commands to assist you in your debugging tasks. There are three ways of entering commands, but not every command can be entered in every way:

For specific information on command syntax, usage, and notation conventions, refer to the *Xpediter/CICS Quick Reference.*

- **Primary Commands** — These commands are entered in the COMMAND field of any screen.
- **Line Commands** — A subset of the primary commands, line commands are typed in the line number area of the display (2.L screen) or in the prefix command field that may precede the data displayed on a line.
- **PF Keys** — PF keys are set by default to the most commonly used commands; they are executed simply by pressing the key. If you wish, you can reset the PF key assignments as part of your user profile. See “PF Key Settings” on page 1-12.

Common Primary Commands

Some commonly used Xpediter/CICS primary commands are:

AFTER

Sets conditional or unconditional breakpoints after the execution of a statement or instruction.

BEFORE

Sets conditional or unconditional breakpoints before the execution of a statement or instruction.

CALC

Performs hexadecimal/decimal calculations and displays the results online.

COUNT

Sets execution counts and gathers test coverage statistics.

DELETE

Removes either a specific object or a dataset record, depending on the screen you access.

EXCLUDE

Excludes specified lines from display. You see only the lines of code in which you are interested. When you step through a program, lines are redisplayed as they are executed so you can see the logic flow of the program. Symbolic label support is provided so that you can exclude a range of lines.

EXIT (PF4)

Transfers to the Exit Session screen.

FIND

Positions the cursor on a specified string. Can be used in conjunction with the EXCLUDE command.

GO (PF12)

Executes program logic by the specified parameter. If no parameters are specified, resumes execution from the current location. GO 5 1 will execute five statements, pausing one second between each statement. GO UNTIL WA-HOURS executes until the value of WA-HOURS changes.

GOTO

Used to reposition execution at another point in the program. Repositions the current execution pointer to the specified statement, offset, or address.

GPREGS

Sets footing options to the general purpose registers.

HELP (PF1)

Displays hypertext online help. HELP COMMANDS displays a list of the available commands. HELP NEWS gives information about the current release. Entering HELP on a particular screen will display information about that screen.

KEEP

Selects the contents of a data item to be viewed in the scrollable and sizeable keep window. To change displayed data, overtype it with a new value.

*LOCATE * (PF6)*

Transfers from any Xpediter screen directly to the Source Listing screen (2.L) or Assembler Break/Abend screen (2.20).

REPEAT

Re-executes the last primary command.

RUNTO

Sets a one-time unconditional breakpoint before the execution of a statement or instruction. GO will also be issued if you are at a trapped transaction.

SET

Changes debugging parameters:

> **KEEPS:** Changes the size of the keep window.
>
> **JUSTIFICATION:** Clips the current source listing so that extraneous data in the source is removed from the display. SET JUSTIFICATION ON clips the lines and suppresses all lines before the first and after the last source line in the display.
>
> **REGS:** Specifies the display format (64 or 32) for the General Purpose registers shown on the "Assembler Break/Abend" screen (2.20) and on the REGISTERS footing. Ignored when CICS is not running on a z/Architecture machine.
>
> **SOURCE:** Provides either a source or break/abend focus.
>
> **FOOT:** Changes the data displayed in the FOOTING.
>
> > **DATA:** Displays hex result of last CALC command.
> >
> > **KEYS:** Displays the PF key settings.
> >
> > **REGS:** Displays current program register contents, assembler instruction, and PSW for break/abend.
> >
> > **SOURCE:** Displays five lines of source.

STATUS: Displays status of current task.

FLOAT: Displays current floating point register contents.

SHOW

Modifies the format of selected screens to display data in different forms.

SKIP

Temporarily bypasses the execution of a statement.

USING

Maps data according to a record format from a COBOL or PL/I program in the file utility.

VERIFY

Displays and, if desired, modifies Assembler object code.

WHEN

Sets conditions for pausing program execution.

Common Line Commands

Some commonly used Xpediter/CICS line commands are:

A (After)

Sets an unconditional breakpoint after a statement.

AC (After Conditional)

Sets a conditional breakpoint after a statement.

B (Before)

Sets an unconditional breakpoint before a statement.

BC (Before Conditional)

Sets a conditional breakpoint before a statement.

C/CC (Count)

Sets an execution analysis/count for a statement or range of statements.

D/DD (Delete)

Deletes Afters, Befores, Counts, Keeps, Skips, conditional Skips, and Verifies.

DH (Delete Hex)

Reverts the display of contents of a kept data name from vertical hex format back to standard format.

GT (Go To)

Repositions the execution pointer (=====>).

H (Hex)

Displays contents of a kept data name in vertical hex format. DH reverts contents back to standard format.

K/KK (Keep)

Selects data names to be kept.

P (Peek)

Transfers to Variable Storage screen (2.3) positioned to the first variable selected.

S/SS (Skip)

Indicates that the selected line or range of lines are to be skipped.

SC (Skip Conditional)

Indicates that the selected line is to be skipped if the specified condition is true.

X/XX (Exclude)

Excludes a line or range of lines.

Z (Runto)

Sets a one-time unconditional breakpoint before the execution of a statement or instruction. GO will also be issued if you are at a trapped transaction.

PF Key Settings

Xpediter/CICS uses PF keys for command shortcuts. For example, to display HELP information about the SHOW command, type SHOW in the COMMAND field and press PF1 (HELP). You can elect to display the current PF key assignments at the bottom of your non-menu screens. To set this option, use the SET FOOT KEYS command.

The default values for the PF keys are shown in Table 1-1. To customize the PF keys to suit your individual needs, use the KEYS primary command.

Table 1-1. Default PF Key Settings

Function	PF Key	Description
HELP	PF1	Provides help information on the function currently in use.
RUNTO CSR	PF2	Generates a runto before breakpoint on the source line containing the cursor. If you are at a trapped task, the GO command will also be issued, and the runto breakpoint will be deleted when it is subsequently taken.
END	PF3	Ends current function and returns to the next higher level screen. For DB2 screens only, returns to previous screen.
=X	PF4	Transfers to the Exit Session screen.
RFIND	PF5	When used with a previously issued FIND command, searches for the next occurrence of the specified data.
LOCATE *	PF6	Directly transfers from any screen to the Source Listing screen (2.L) or Assembler Break/Abend screen (2.20).
UP	PF7	Scrolls up through the data portion of a screen or the Help facility.
DOWN	PF8	Scrolls down through the data portion of a screen or the Help facility.
GO 1	PF9	Resumes program execution for one instruction or statement, then halts the program.
LEFT	PF10	Scrolls the data portion of the screen to view data to the left of the current display. On the Memory Display screens (2.2 and 9.2), PF10 acts as the PREV command.
RIGHT	PF11	Scrolls the data portion of the screen to view data to the right of the current display. On the Memory Display screens (2.2 and 9.2), PF11 acts as the NEXT command.
GO	PF12	Steps through program logic by the specified parameters. If no parameters are specified, resumes program from current location. This key is available in all situations where resuming is allowed.

Table 1-1. Default PF Key Settings

Function	PF Key	Description
MENU	PF13	Ends the current Xpediter/CICS function and transfers to the highest level System Menu. From the Help facility or the Primary Menu, the copyright and trade secret notices are displayed.
MEMORY	PF14	Transfers to the Memory Display screens (2.2). On the Primary Menu, PF14 displays the copyright and trade secret notices.
SELECT	PF15	Selects a function from a menu or selection screen, or lists the sixteen most recently accessed addresses from the Memory Display screens (2.2 and 9.2).
WS, DS, or VS	PF16	Transfers to the Working Storage screen (2.3) for COBOL, the Defined Storage screen (2.3) for Assembler, or the Variable Storage screen (2.3) for PL/I and C.
=2.4	PF17	Transfers to the Program Trace screen (2.4).
=2.8	PF18	Transfers to the Last 3270 screen (2.8).
UP MAX	PF19	Scrolls up the maximum amount possible.
DOWN MAX	PF20	Scrolls down the maximum amount possible.
FILE	PF21	Transfers to the File Utility Menu (5).
DLEFT	PF22	Scrolls the data in the keep window to view data to the left of the current display.
DRIGHT	PF23	Scrolls the data in the keep wondow to view data to the right of the current display.
RETRIEVE	PF24	Displays the last command entered in the COMMAND field, allowing it to be changed or reissued.

Update Password Security

The Update Security field provides security from memory updates on selected screens. When this security is implemented by the Xpediter/CICS system administrator, unauthorized users are prevented from making updates to the following screens:

- **2.2** Memory Display (XPED/XPRT)
- **2.D** CICS DSECTs (XPED/XPRT)
- **5.1.3** Edit CICS Dataset Record
- **5.2.3** Edit Queued Record
- **5.3.2** Edit Transient Data Queue Record
- **5.4.4** Edit DL/1 Segment
- **5.5.5** DB2 Edit Result Table Row
- **5.5.6** DB2 Edit Composite Column
- **5.6.3** Update MQ Queue Message
- **9.2** Memory Display (XPSP)
- **9.D** CICS DSECTs (XPSP).

When this facility is enabled, the following field is displayed on the screen when using a given transaction:

```
UPDATE PASSWORD: xxxxxxxx
```

Where **xxxxxxxx** is a non-displayed field used for entering the password.

To update memory displayed on the screen, enter the password and modify the data to be changed *before* pressing Enter or a PF key. If no password is entered or the password is invalid, all modifications to the data area of the screen are ignored. If memory is not updated, you receive a message indicating why the update was bypassed.

Chapter 2.
Getting Started

This chapter gives you the basic information needed to start using Xpediter/CICS. It discusses the steps necessary to prepare your program for execution, including compiling, processing, and loading a new copy. The chapter also discusses how to access, navigate through, and exit Xpediter/CICS.

Take a few minutes to read this chapter before starting the exercises in this guide. It answers many common questions.

Preparing a Program for Execution

It is possible to debug a program without any preparation other than entering XPED and your program name. You work at the operation-code level, set breakpoints and examine program storage using offsets. This process is called *sourceless debugging*. Sourceless debugging is valuable when working with programs that can't be recompiled, such as vendor packages or production versions of programs.

For examples of sourceless debugging, refer to Chapter 4, “Debugging Applications Without Source Code”.

The typical method of debugging programs, however, is to use Xpediter's *source-level support* for PL/I.

To take advantage of this support, you must run your PL/I programs through the PL/I language processor, which provides you with the following functionality during a debugging session:

- Set one or more breakpoints to stop execution at any statement. Breakpoints may be:
  - *Unconditional:* Halts program execution at the selected statement.
  - *Conditional:* Halts program execution only if the specified condition is met.
- Examine program storage by data name.
- Select data fields for viewing during program execution.
- Resume execution at any statement.
- Change the execution point by statement number.
- Examine the statement-level trace during or after execution.

Note: Certain PL/I verbs—DO, for example—may compile without generating any executable code. If you use the Source Listing screen (2.L) to set a breakpoint on a statement containing one of these verbs, the breakpoint will actually be placed on the next executable line. If a conditional breakpoint already exists on that line, Xpediter will disallow the newer breakpoint and issue an error message.

Language processing lets you work with the code in the way you are accustomed to seeing it: at the source level. The next section discusses the Compuware language processor and how to get source-level support.

The Compuware Language Processor

The Compuware language processor is a set of programs that captures information about a compiler listing and stores it in a source listing file, also known as a DDIO file. You have the option to use either the preprocessor or the postprocessor. As certain information is not available from the compiler listing, Xpediter/CICS recommends that you use the preprocessor when installing Release 9.1 because it gathers additional information and provides you with the following benefits:

- **Simplified JCL:** While the postprocessor requires that the user add a step after the compile step, the preprocessor requires only that the EXEC name be altered and a CWPDDIO DD card and CWPPRMO DD card be added.
- **Automated print options:** The postprocessor requires that certain compiler options be specified in order to print all needed sections of the compiler listing. The preprocessor can automatically pass the required options to the compiler.
- **Capturing of suppressed source code:** When PL/I %NOPRINT is used, sections of source code can be suppressed from the compiler listing. The preprocessor can capture this information from the compiler before the data is suppressed from the listing. This provides improved debugging under CICS.

To enable source-level support, the language processor must be installed. The most up to date version of Compuware Shared Services is furnished with Xpediter/CICS and is usually installed as part of the Xpediter installation process. For information on how to install the Compuware Shared Services language processor, refer to the *Xpediter/CICS Installation Guide* and the *Enterprise Common Components Installation and Customization Guide*.

During a debugging session, Xpediter/CICS searches the source listing file for an entry that matches your program name. It also checks the compile date and time. For Assembler and PL/I, if no compile date and time are found in the load module, Xpediter checks the ASMSTMP or PL1STMP parameter in the global table. If the parameter is set to YES (the default), the NO SOURCE AVAILABLE message is displayed. Press the help PF key (default PF1) for specific information. If the parameter is set to NO, Xpediter bypasses timestamp validation and continues. For more information on these parameters and how to create a timestamp in Assembler and PL/I programs, refer to Chapter 5, "Global Parameters Table" in the *Xpediter/CICS Installation Guide*.

Using the NEWCOPY Function

Whenever a program is recompiled, a new copy of the program must be loaded in CICS before attempting to test the recompiled version with Xpediter/CICS. The Xpediter NEWC transaction should be used instead of the CEMT transactions to load a new copy of a program into CICS.

The NEWC transaction loads a new copy and resets generic breakpoints. Generic breakpoints are those set without reference to a specific statement number or offset, such as those created by the commands BEFORE ALL PARA and AFTER ALL EXEC. Explicit, non-generic breakpoints are set with commands such as BEFORE 100 and AFTER +24 — as well as the line commands B, A, and C — and are deleted from the new copy of the program created with the NEWC transaction.

To use the NEWC transaction, type **NEWC** followed by the program name on a blank CICS screen. If the NEWCOPY is successful, the following message is displayed:

```
NEWC CWDEMCB2
MXDNC0001I 01 Jan 2006 09:00:00 - SYSID=C123 - APPLID=CICSC123
 Program 'CWDEMCB2' newcopy successful  Len(00021E8)..
```

The program name in the first line of the message may be overtyped to issue an additional NEWCOPY request.

If the program is in use or the program name is entered incorrectly, error messages are returned. Any authorized user in the region can NEWCOPY a program; if the program is in use, an error message is issued, but the program is not disabled.

For more information on the NEWCOPY function, refer to the *Xpediter/CICS Reference Manual*.

Accessing Xpediter/CICS

There are three ways to access Xpediter/CICS:

- Basic access
- Fast-path access to the Source Listing screen (2.L)
- Direct access to selected screens.

Any of the three transactions, XPED, XPRT, or XPSP, can be used with each method. XPED and XPRT automatically issue the abend trap facility to trap any potential abends in your program before they can actually occur.

Xpediter can also be configured to run in any of three restricted modes of operation:

- Diagnosis Mode
- Utilities Mode
- Diagnosis/Utilities Mode.

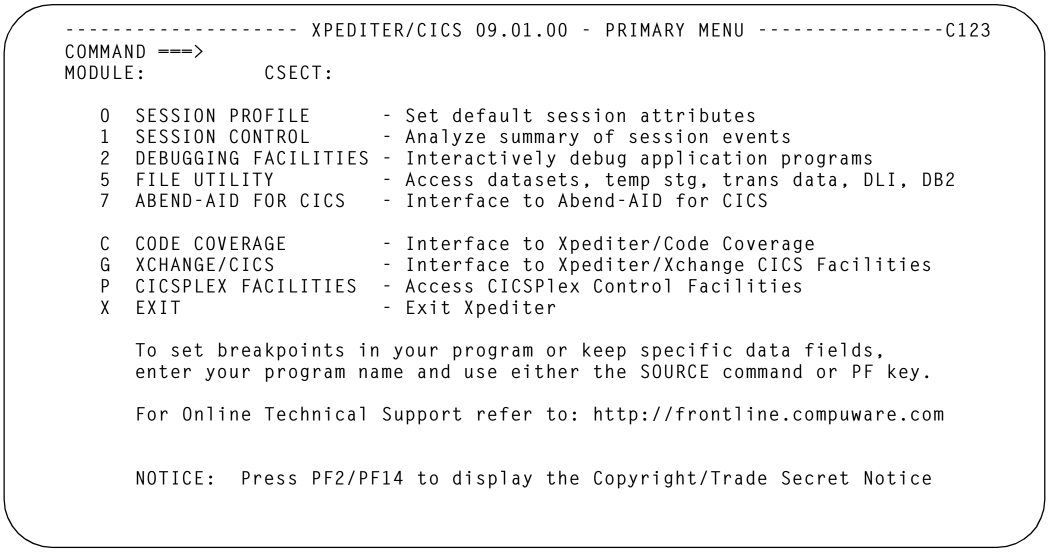
For more information, see the *Xpediter/CICS Reference Manual*.

Note: All descriptions in this manual, unless otherwise noted, pertain to Xpediter's standard, non-restricted operating mode.

Basic Access

The easiest way to access Xpediter/CICS is by entering XPED, XPRT, or XPSP on a blank CICS screen and pressing Enter. XPED and XPRT take you to the Primary Menu shown in Figure 2-1. XPSP takes you to a similar Primary Menu with additional functions for system programmers. The Primary Menu will only display certain choices if the corresponding Compuware product or functionality—such as Xpediter/*Xchange*, Xpediter/Code Coverage, or CICSPlex support—is installed and active in the CICS region.

Figure 2-1. Primary Menu



```
------------------- XPEDITER/CICS 09.01.00 - PRIMARY MENU ----------------C123
COMMAND ===>
MODULE:          CSECT:

   0  SESSION PROFILE      - Set default session attributes
   1  SESSION CONTROL      - Analyze summary of session events
   2  DEBUGGING FACILITIES - Interactively debug application programs
   5  FILE UTILITY         - Access datasets, temp stg, trans data, DLI, DB2
   7  ABEND-AID FOR CICS   - Interface to Abend-AID for CICS

   C  CODE COVERAGE        - Interface to Xpediter/Code Coverage
   G  XCHANGE/CICS         - Interface to Xpediter/Xchange CICS Facilities
   P  CICSPLEX FACILITIES  - Access CICSPlex Control Facilities
   X  EXIT                 - Exit Xpediter

      To set breakpoints in your program or keep specific data fields,
      enter your program name and use either the SOURCE command or PF key.

      For Online Technical Support refer to: http://frontline.compuware.com


      NOTICE:  Press PF2/PF14 to display the Copyright/Trade Secret Notice
```

From the Primary Menu, you can:

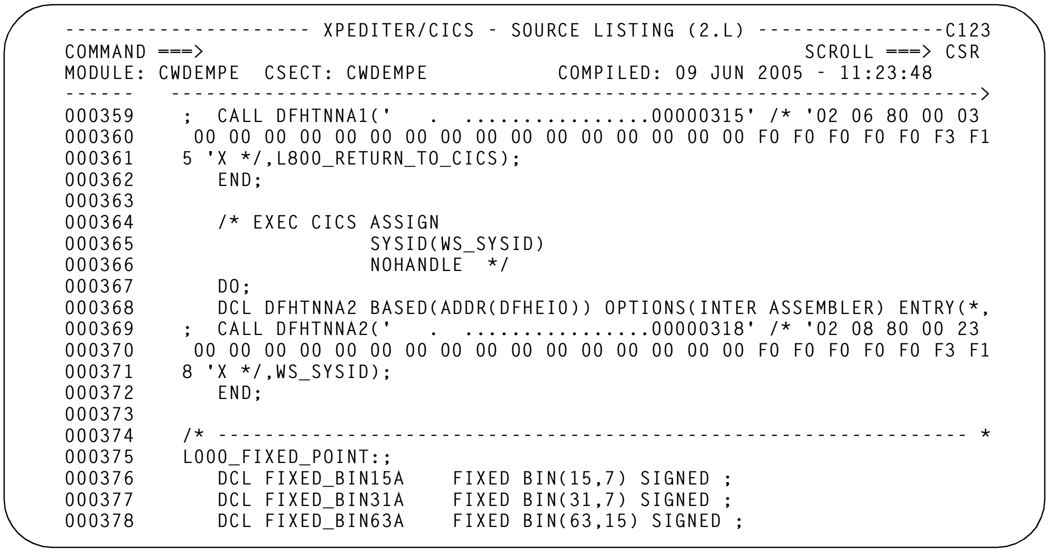
- Transfer to any of the options shown on the menu by typing the appropriate number and pressing Enter.
- Type a program name in the MODULE field and press the LOCATE * PF key (default PF6) to transfer to the Source Listing screen (2.L).
- Press Clear to return to CICS to start your application.

Fast-Path Access to the Source Listing screen (2.L)

To directly access the Source Listing screen (2.L), enter an Xpediter/CICS transaction, followed by the load module name. This can be done, for example, by typing **XPED CWDEMPE** and pressing Enter. The Source Listing screen (2.L) is displayed for the Xpediter/CICS PL/I demonstration program CWDEMPE. You can also specify a load module and CSECT combination separated by a colon (:).

From here you can set breakpoints in your programs, select variables to keep for viewing, set up an execution analysis, specify statements to be skipped, or transfer to another screen.

Figure 2-2. Accessing the Source Listing Screen (2.L) Using the Fast-Path Method



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48
------   ------------------------------------------------------------------->
000359    ;  CALL DFHTNNA1('   .  ................00000315' /* '02 06 80 00 03
000360     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0 F0 F0 F0 F0 F3 F1
000361    5 'X */,L800_RETURN_TO_CICS);
000362       END;
000363
000364       /* EXEC CICS ASSIGN
000365                    SYSID(WS_SYSID)
000366                    NOHANDLE  */
000367       DO;
000368       DCL DFHTNNA2 BASED(ADDR(DFHEIO)) OPTIONS(INTER ASSEMBLER) ENTRY(*,
000369    ;  CALL DFHTNNA2('   .  ................00000318' /* '02 08 80 00 23
000370     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0 F0 F0 F0 F0 F3 F1
000371    8 'X */,WS_SYSID);
000372       END;
000373
000374    /* ------------------------------------------------------------------ *
000375    L000_FIXED_POINT:;
000376       DCL FIXED_BIN15A    FIXED BIN(15,7) SIGNED ;
000377       DCL FIXED_BIN31A    FIXED BIN(31,7) SIGNED ;
000378       DCL FIXED_BIN63A    FIXED BIN(63,15) SIGNED ;
```

Direct Access to Selected Screens

When you know the screen that you want to access, enter an Xpediter/CICS transaction, followed by the screen ID. For example, type **XPED 5** and press Enter to display the File Utility Menu (5).

Navigating Through Xpediter/CICS

Xpediter/CICS uses techniques similar to ISPF to transfer from screen to screen. The following methods are used to navigate through Xpediter/CICS:

- Type one of the screen numbers from a displayed menu in the COMMAND field and press Enter.
- Move the cursor next to the desired screen number on a menu, type **S**, and press Enter.
- Transfer between screens by typing = followed by the screen ID. For example, type =**5.1** in the COMMAND field and press Enter to transfer to the CICS Datasets Menu (5.1).
- Use PF keys to transfer to commonly used screens. The PF keys and defaults for the corresponding screens are as follows:

PF6	LOCATE * (2.L or 2.20)
PF13	Primary Menu
PF14	Memory Display screen (2.2)
PF16	Defined Storage screen (2.3)
PF17	Program Trace screen (2.4)
PF21	File Utility Menu (5).

- Access the Source Listing screen (2.L) from any product screen with the LOCATE ***** primary command. For more information, refer to the *Xpediter/CICS Reference Manual*.
- Exit Xpediter/CICS by typing =**X** in the COMMAND field and pressing Enter.

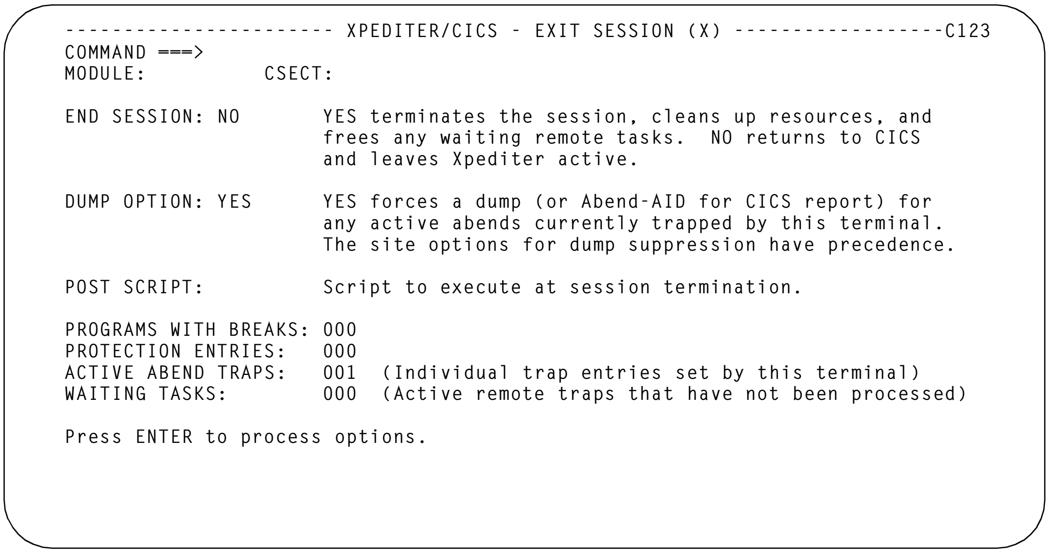
Exiting Xpediter/CICS and Ending a Debugging Session

All of the breakpoints, skips, counts, and keeps you set are associated with your terminal. When you finish testing, it is very important that you end your debugging session. This frees up any resources that may have been used during the session (including abend traps which intercept programs when they abend) and releases all breakpoints, keeps, skips, and counts from your programs.

Ending a Session

1. Type **=X** in the COMMAND field on any Xpediter/CICS screen. If you are not currently in Xpediter/CICS, type **XPED X** on a blank CICS screen. Press Enter. The Exit Session screen (X) appears as shown in Figure 2-3.

Figure 2-3. Exit Session Screen (X)



```
---------------------- XPEDITER/CICS - EXIT SESSION (X) -----------------C123
COMMAND ===>
MODULE:          CSECT:

END SESSION: NO       YES terminates the session, cleans up resources, and
                      frees any waiting remote tasks.  NO returns to CICS
                      and leaves Xpediter active.

DUMP OPTION: YES      YES forces a dump (or Abend-AID for CICS report) for
                      any active abends currently trapped by this terminal.
                      The site options for dump suppression have precedence.

POST SCRIPT:          Script to execute at session termination.

PROGRAMS WITH BREAKS: 000
PROTECTION ENTRIES:   000
ACTIVE ABEND TRAPS:   001  (Individual trap entries set by this terminal)
WAITING TASKS:        000  (Active remote traps that have not been processed)

Press ENTER to process options.
```

2. Type **Y** (for YES) in the END SESSION field.
3. If you had trapped a transaction and would like a dump, type **Y** in the DUMP OPTION field.
4. If you would like to execute a user or system script, type the script member name in the POST SCRIPT field.
5. Press Enter. Xpediter/CICS displays the message **XPEDITER/CICS SESSION TERMINATED - SYSID=C123** to show that any resources used during the debugging session are released.

You may also exit Xpediter directly, bypassing the Exit Session screen (X), by typing **XPND** on a blank CICS screen.

The Resource Summary screen (1.P) can be used to release breakpoints set from the local terminal, and the XPSP transaction's Resource Summary screen (9.P) can be used to release breakpoints from programs in the entire CICS region. This is especially helpful if your site uses autoinstall terminals and you sign off CICS without ending an Xpediter/CICS session. For more information, see the screen descriptions for the Resource Summary screens (1.P and 9.P) in the *Xpediter/CICS Reference Manual*.

Usage Considerations for PL/I

The following considerations apply when debugging PL/I programs:

- When multiple generations of controlled variables are allocated, only the most current generation is displayed in the keep window and on the Variable Storage screen (2.3).
- If a procedure is called recursively, the procedure's display of automatic variables in the keep window and on the Variable Storage screen (2.3) is limited to those variables for the most current invocation of that procedure.

The following usage considerations apply to the PL/I language processor in the following areas:

- Declaration of variables
- Compiler/run-time options
- Programming techniques.

For a complete list of these restrictions, refer to the *Compuware Shared Services User/Reference Guide*.

The following conditions are not supported by the PL/I language processor:

- Variables with adjustable length that are defined in automatic storage.

```
DCL ARRAY(X,Y) CHAR(1);
```

- Multiple programs that are compiled together.

```
%PROCESS...
PROC1:PROC OPTIONS(MAIN);
    :
%PROCESS...
PROC2: PROC OPTIONS(MAIN);
```

- LANGUAGE run-time options.

```
DCL PLIXOPT CHAR(12) VAR INIT(LANGUAGE(JA)) STATIC EXTERNAL
```

- Mixed case is not fully supported.

```
DeClArE VaRa ChAr (1);
```

- An array element used as a base pointer.

```
DCL BADDR(5) PTR;
DCL VARA BASED(BADDR(3));
```

- An expression used as a base pointer.

```
DCL VARA BASED(BADDR+8);
```

- AREA variables ≥ 32K.

```
DCL VARA AREA(32768);
```

In addition, the following rules apply to %INCLUDE and %NOPRINT statements:

- The character strings %INCLUDE and %NOPRINT cannot be split and continued on the next line.

```
                              %NOP
RINT;
```

- For %INCLUDE and %NOPRINT, the language processor will recognize a maximum of one space between the % character and the command that follows (i.e., % INCLUDE or % NOPRINT).

  ```
  %     INCLUDE X;
  ```

Note: Other product-specific restrictions may also apply when using the PL/I language processor. Consult your product documentation for more information.

Chapter 3. Testing a PL/I Program

This chapter demonstrates how to test a PL/I application program, first without setting breakpoints, and then by setting breakpoints and stepping through the code. You will use the XPED transaction to test an application transaction (XPLE) and fix an abend. XPLE is the sample demonstration transaction shipped with Xpediter/CICS. It is a simple employee payroll transaction that executes the PL/I program CWDEMPE. The XPLE transaction is used throughout this guide to cause several types of abends.

If you have questions about which screen to use for a function, the *Xpediter/CICS Quick Reference* contains a list of screen IDs and titles, default PF key settings, and commands.

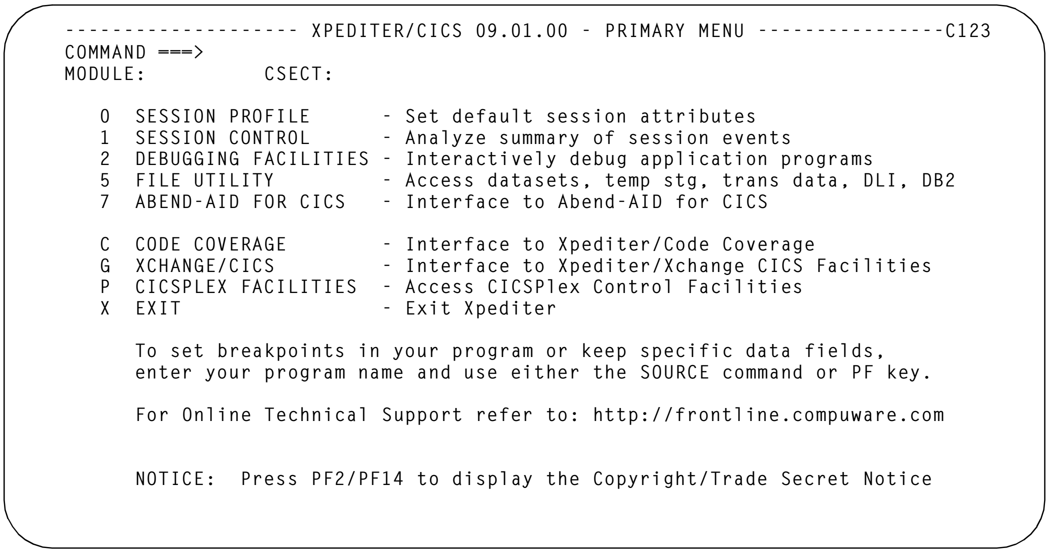
Note: The demonstrations in this chapter should be performed in Xpediter's standard operating mode — ***not*** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

Testing without Breakpoints

1. On a blank CICS screen, type **XPED** and press Enter.

   This activates Xpediter/CICS, sets the abend trap option, and displays the Primary Menu as shown in Figure 3-1. The Primary Menu will only display certain choices if the corresponding Compuware product or functionality—such as Xpediter/*Xchange*, Xpediter/Code Coverage, or CICSPlex support—is installed and active in the CICS region.

Figure 3-1. Primary Menu (XPED/XPRT)



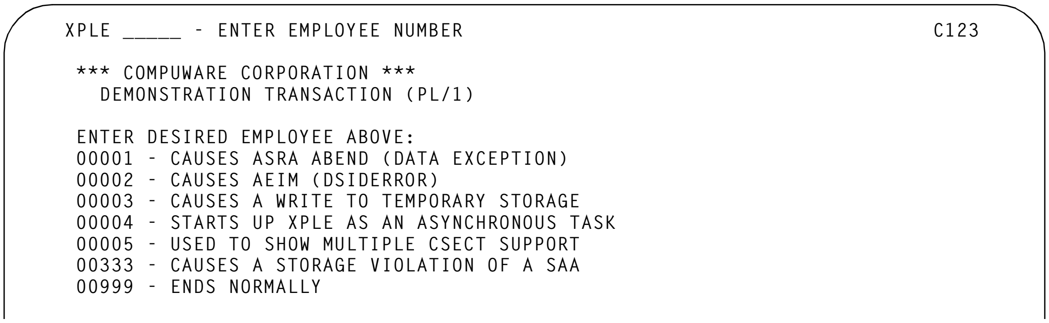
```
-------------------- XPEDITER/CICS 09.01.00 - PRIMARY MENU ----------------C123
COMMAND ===>
MODULE:          CSECT:

   0  SESSION PROFILE      - Set default session attributes
   1  SESSION CONTROL      - Analyze summary of session events
   2  DEBUGGING FACILITIES - Interactively debug application programs
   5  FILE UTILITY         - Access datasets, temp stg, trans data, DLI, DB2
   7  ABEND-AID FOR CICS   - Interface to Abend-AID for CICS

   C  CODE COVERAGE        - Interface to Xpediter/Code Coverage
   G  XCHANGE/CICS         - Interface to Xpediter/Xchange CICS Facilities
   P  CICSPLEX FACILITIES  - Access CICSPlex Control Facilities
   X  EXIT                 - Exit Xpediter

      To set breakpoints in your program or keep specific data fields,
      enter your program name and use either the SOURCE command or PF key.

      For Online Technical Support refer to: http://frontline.compuware.com


      NOTICE:  Press PF2/PF14 to display the Copyright/Trade Secret Notice
```

   Xpediter/CICS is now turned on and ready to intercept any abends that may occur.

2. To turn on the trace facility, type **SET TRACE ON** on the COMMAND line and press Enter.

3. Press Clear to return to CICS to start your test.
4. Type **XPLE** and press Enter. The Demonstration Transaction screen appears as shown in Figure 3-2.

Figure 3-2. Demonstration Transaction Screen

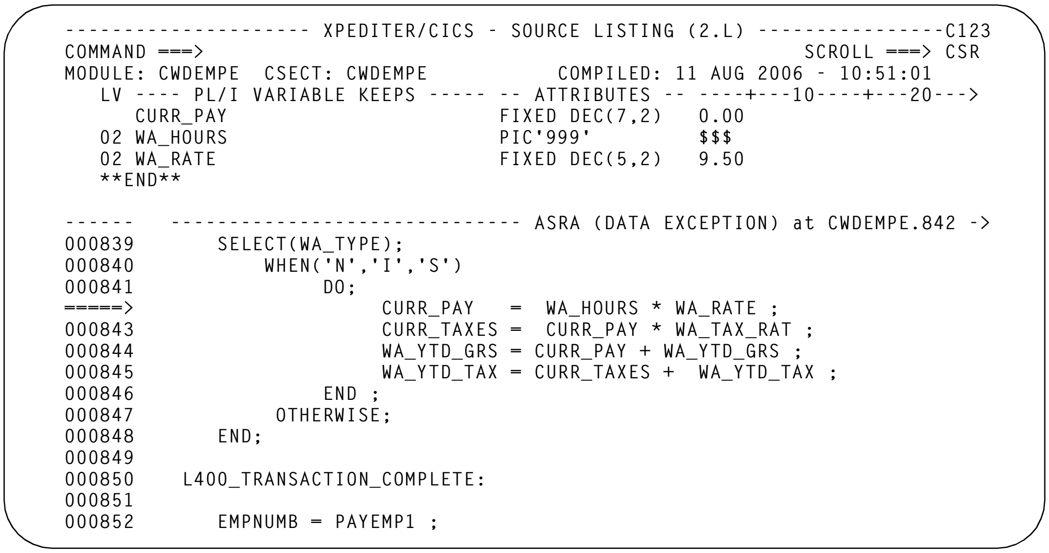


```
XPLE _____ - ENTER EMPLOYEE NUMBER                                        C123

 *** COMPUWARE CORPORATION ***
   DEMONSTRATION TRANSACTION (PL/1)

 ENTER DESIRED EMPLOYEE ABOVE:
 00001 - CAUSES ASRA ABEND (DATA EXCEPTION)
 00002 - CAUSES AEIM (DSIDERROR)
 00003 - CAUSES A WRITE TO TEMPORARY STORAGE
 00004 - STARTS UP XPLE AS AN ASYNCHRONOUS TASK
 00005 - USED TO SHOW MULTIPLE CSECT SUPPORT
 00333 - CAUSES A STORAGE VIOLATION OF A SAA
 00999 - ENDS NORMALLY
```

5. To cause an ASRA abend, type **00001** and press Enter. The CWDEMPE demo program is intercepted when the abend occurs, and the Source Listing screen (2.L) is displayed as shown in Figure 3-3. This gives you the opportunity to fix the problem, re-execute the statement, and continue the test.

Note: The line numbers shown in this guide may vary from those seen during actual program execution.

Figure 3-3. Source Listing Screen (2.L) Showing an ASRA



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ---------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 11 AUG 2006 - 10:51:01
   LV ---- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
      CURR_PAY                        FIXED DEC(7,2)   0.00
   02 WA_HOURS                        PIC'999'         $$$
   02 WA_RATE                         FIXED DEC(5,2)   9.50
   **END**

------   ------------------------------ ASRA (DATA EXCEPTION) at CWDEMPE.842 ->
000839       SELECT(WA_TYPE);
000840           WHEN('N','I','S')
000841                DO;
=====>                     CURR_PAY   =  WA_HOURS * WA_RATE ;
000843                     CURR_TAXES =  CURR_PAY * WA_TAX_RAT ;
000844                     WA_YTD_GRS = CURR_PAY + WA_YTD_GRS ;
000845                     WA_YTD_TAX = CURR_TAXES +  WA_YTD_TAX ;
000846                END ;
000847            OTHERWISE;
000848       END;
000849
000850    L400_TRANSACTION_COMPLETE:
000851
000852       EMPNUMB = PAYEMP1 ;
```

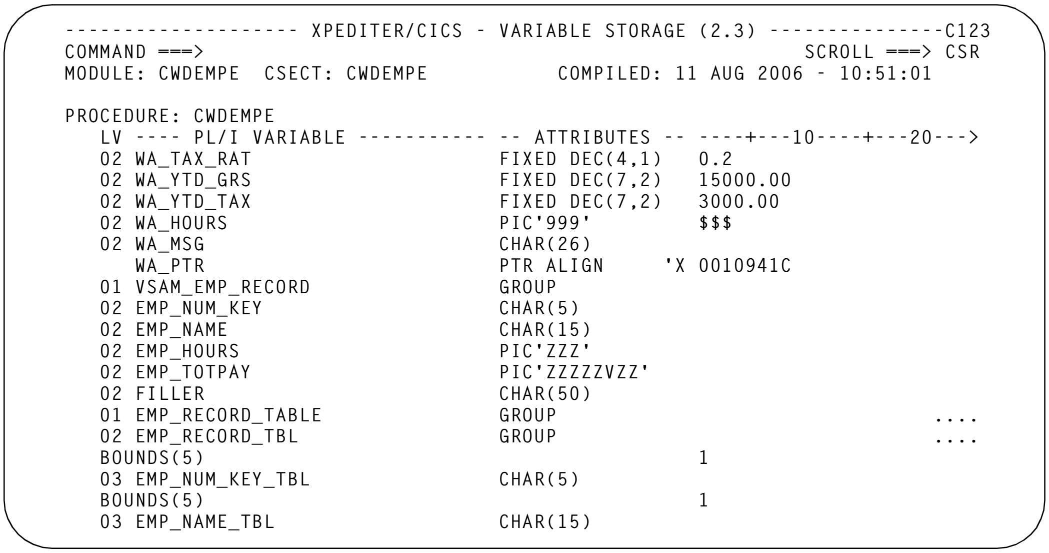
The module and CSECT names, as well as the compile date and time, of the executing program are displayed at the top of the screen to verify that the correct program is running. The keep window appears next. Notice that all data items from the current statement automatically appear. These are called automatic keeps. Scroll through this window by positioning the cursor and using PF keys to scroll up, down, left, and right.

The status line is displayed after the keep window. The message indicates that an ASRA abend caused by a data exception has been intercepted at statement 842 in CWDEMPE.

The source code follows the STATUS line. Scroll through this section by positioning the cursor anywhere on the screen outside the keep window. An arrow indicates the current statement.

6. The contents of other data items can be checked by using the PEEK primary command. Type **PEEK WA_TAX_RAT** in the COMMAND field and press Enter. Xpediter/CICS will transfer to the Variable Storage screen (2.3) with the contents of WA_TAX_RAT positioned to the top of the screen as shown in Figure 3-4.

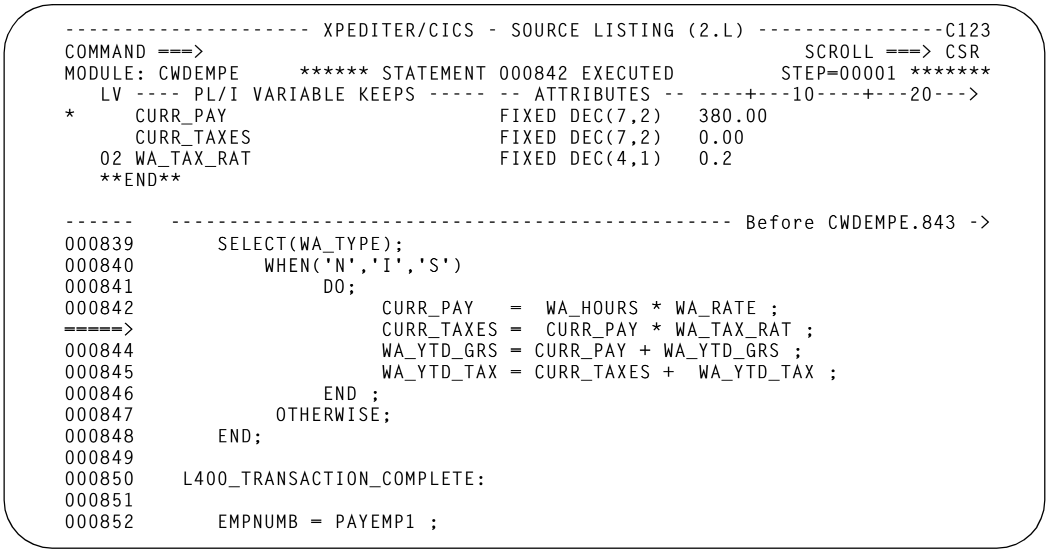
Figure 3-4. Variable Storage Screen (2.3) Accessed with PEEK Command



```
-------------------- XPEDITER/CICS - VARIABLE STORAGE (2.3) ---------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 11 AUG 2006 - 10:51:01

PROCEDURE: CWDEMPE
   LV ---- PL/I VARIABLE ----------- -- ATTRIBUTES -- ----+---10----+---20--->
   02 WA_TAX_RAT                     FIXED DEC(4,1)   0.2
   02 WA_YTD_GRS                     FIXED DEC(7,2)   15000.00
   02 WA_YTD_TAX                     FIXED DEC(7,2)   3000.00
   02 WA_HOURS                       PIC'999'         $$$
   02 WA_MSG                         CHAR(26)
      WA_PTR                         PTR ALIGN     'X 0010941C
   01 VSAM_EMP_RECORD                GROUP
   02 EMP_NUM_KEY                    CHAR(5)
   02 EMP_NAME                       CHAR(15)
   02 EMP_HOURS                      PIC'ZZZ'
   02 EMP_TOTPAY                     PIC'ZZZZZVZZ'
   02 FILLER                         CHAR(50)
   01 EMP_RECORD_TABLE               GROUP                                ....
   02 EMP_RECORD_TBL                 GROUP                                ....
   BOUNDS(5)                                          1
   03 EMP_NUM_KEY_TBL                CHAR(5)
   BOUNDS(5)                                          1
   03 EMP_NAME_TBL                   CHAR(15)
```

7. Press PF3 (END) to return to the Source Listing screen (2.L).
8. Note the value of WA_HOURS. The bad data ($$$) in this field is causing the ASRA. To change it, position the cursor on the bad data, type **040**, and press Enter.
9. Press PF9 (GO 1) to execute one statement. Notice that the values of the data fields in the keep window change (Figure 3-5).

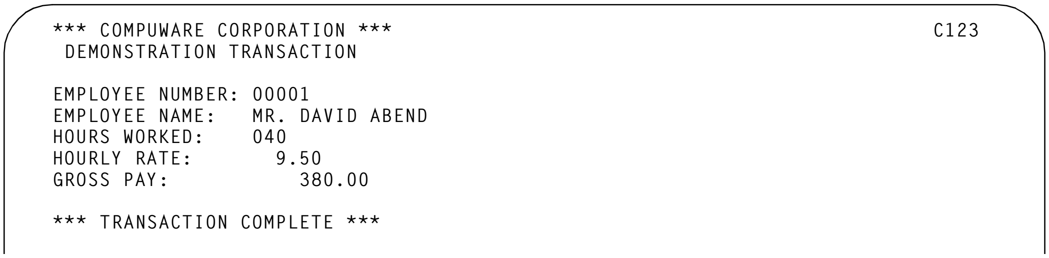
Figure 3-5. After GO 1 on the Source Listing Screen (2.L)



```
--------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE: CWDEMPE     ****** STATEMENT 000842 EXECUTED        STEP=00001 *******
   LV ---- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
*     CURR_PAY                       FIXED DEC(7,2)   380.00
      CURR_TAXES                     FIXED DEC(7,2)   0.00
   02 WA_TAX_RAT                     FIXED DEC(4,1)   0.2
   **END**

------   --------------------------------------------- Before CWDEMPE.843 ->
000839       SELECT(WA_TYPE);
000840           WHEN('N','I','S')
000841                DO;
000842                     CURR_PAY   =  WA_HOURS * WA_RATE ;
=====>                     CURR_TAXES =  CURR_PAY * WA_TAX_RAT ;
000844                     WA_YTD_GRS = CURR_PAY + WA_YTD_GRS ;
000845                     WA_YTD_TAX = CURR_TAXES +  WA_YTD_TAX ;
000846                END ;
000847            OTHERWISE;
000848       END;
000849
000850    L400_TRANSACTION_COMPLETE:
000851
000852       EMPNUMB = PAYEMP1 ;
```

10. Press PF12 (GO), which continues the test. If any other abends occur, Xpediter/CICS intercepts them. Otherwise, the transaction completes, and the Demonstration Transaction screen appears as shown in Figure 3-6.

Figure 3-6. Demonstration Transaction Screen



```
*** COMPUWARE CORPORATION ***                                           C123
 DEMONSTRATION TRANSACTION

EMPLOYEE NUMBER: 00001
EMPLOYEE NAME:   MR. DAVID ABEND
HOURS WORKED:    040
HOURLY RATE:       9.50
GROSS PAY:           380.00

*** TRANSACTION COMPLETE ***
```

11. Remember to end the session as described in “Exiting Xpediter/CICS and Ending a Debugging Session” on page 2-6.

In this example, an ASRA occurred because WA_HOURS contained bad data. In the next example, we will find out how the bad data got there.

Viewing Source

1. Type **XPED CWDEMPE** on a blank CICS screen and press Enter. This fastpath access method lets you immediately see source for a specific program.

   During execution, Xpediter/CICS automatically displays the data names in the current statement. In addition, you can select any number of data names to be displayed during execution. These are called explicit keeps. Xpediter/CICS displays these fields in the keep window area of the Source Listing screen (2.L) when an abend or a breakpoint occurs. This feature allows you to monitor data names and modify their values during program execution.

   A feature called Intelligent Autokeeps is an extension to Xpediter's autokeep facility. This feature is enabled by default, but can be disabled in your individual profile settings. With Intelligent Autokeeps enabled, if an autokeep variable could be modified by the execution of the current statement, it will be redisplayed in the keep window when you step to the next statement. This *intellikeep* can often eliminate the need to set, then later remove, an explicit keep. The Intelligent Autokeeps feature will not display an autokeep if a duplicate explicit keep has been set.

   Note: For a more complete explanation of autokeeps and the Intelligent Autokeeps feature, Compuware encourages you to use the HELP AUTOKEEPS and HELP INTELLIKEEPS commands built into Xpediter/CICS. These Help topics provide highly detailed information and examples, as well as performing real-time evaluations to point out any restrictions you might encounter while attempting to use these facilities.

   In the previous example, an abend occurred because the field WA_HOURS contained invalid data. By setting a keep on this field, you can view it to monitor its value and check its effect on the program execution.

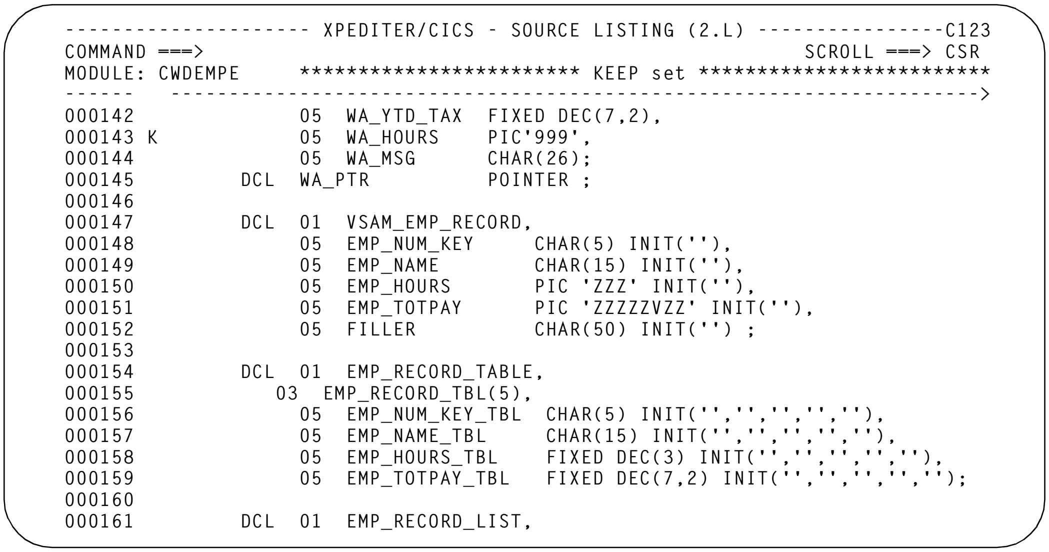
2. To turn on the trace facility, type **SET TRACE ON** on the COMMAND line and press Enter.

3. Type **FIND FIRST WA_HOURS** on the COMMAND line and press Enter. The display is positioned to the declarative (DCL) for WA_HOURS.

4. Type the **K** (Keep) line command to the left of WA_HOURS and press Enter. The value of WA_HOURS will appear in the keep window when you execute the program.

Whenever a keep is set, a **K** is placed on the line where the data name is defined (Figure 3-7).

Figure 3-7. Selecting a Data Name to Keep on the Source Listing Screen (2.L)



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE     ************************ KEEP set ************************
------   --------------------------------------------------------------------->
000142              05  WA_YTD_TAX  FIXED DEC(7,2),
000143 K            05  WA_HOURS    PIC'999',
000144              05  WA_MSG      CHAR(26);
000145         DCL  WA_PTR          POINTER ;
000146
000147         DCL  01  VSAM_EMP_RECORD,
000148              05  EMP_NUM_KEY     CHAR(5) INIT(''),
000149              05  EMP_NAME        CHAR(15) INIT(''),
000150              05  EMP_HOURS       PIC 'ZZZ' INIT(''),
000151              05  EMP_TOTPAY      PIC 'ZZZZZVZZ' INIT(''),
000152              05  FILLER          CHAR(50) INIT('') ;
000153
000154         DCL  01  EMP_RECORD_TABLE,
000155            03  EMP_RECORD_TBL(5),
000156              05  EMP_NUM_KEY_TBL  CHAR(5) INIT('','','','',''),
000157              05  EMP_NAME_TBL     CHAR(15) INIT('','','','',''),
000158              05  EMP_HOURS_TBL    FIXED DEC(3) INIT('','','','',''),
000159              05  EMP_TOTPAY_TBL   FIXED DEC(7,2) INIT('','','','','');
000160
000161         DCL  01  EMP_RECORD_LIST,
```

Setting Program Breakpoints

Next we will set a breakpoint at the beginning of the program so that we can gain control and see the initializaed value of WA_HOURS.

Breakpoints are set to stop execution of a program. They are set at any executable verb and can be set either before or after the statement is executed. There are two different types:

- **Unconditional Breakpoints:** Halt program execution at the selected statement. Intercepts, described in "Intercept Summary (1.7)" on page 3-11, are a subset of unconditional breakpoints. Runtos are one-time before breakpoints. Setting a runto breakpoint also issues the GO command if a task has been trapped at a break or stop. After the runto breakpoint has been taken, it is deleted.
- **Conditional Breakpoints:** Halt program execution only when the specified condition is met.

Note: Certain PL/I verbs—DO, for example—may compile without generating any executable code. If you use the Source Listing screen (2.L) to set a breakpoint on a statement containing one of these verbs, the breakpoint will actually be placed on the next executable line. If a conditional breakpoint already exists on that line, Xpediter will disallow the newer breakpoint and issue an error message.

In this case we'll set a conditional breakpoint on the abending statement and an unconditional breakpoint at the beginning of the program.

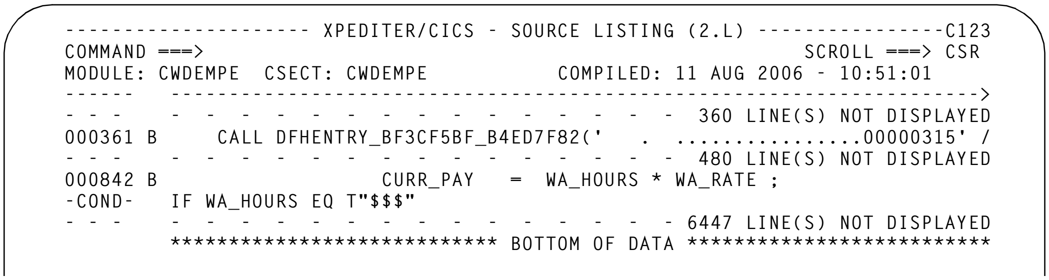
1. Press PF5 (RFIND) twice. The display is positioned to the next occurrence of WA_HOURS, where the abend was intercepted in the first example of this chapter.
2. Type **B** on the statement number and press Enter. This sets an unconditional breakpoint before the statement, meaning the program will pause before this statement is executed. It can easily be changed to a conditional breakpoint.
3. Type **BC** on the statement number and press Enter. A **- COND - IF** line opens under the statement.

4. Type **WA_HOURS EQ "$$$"** next to the IF and press Enter. This sets a conditional breakpoint. This breakpoint is only taken when the value of WA_HOURS is $$$.
5. Type **K3** on the statement number and press Enter. This sets an explicit keep on the third variable in the line (WA_RATE).

   The next command demonstrates setting breakpoints from the command line.
6. Type **BEFORE 0** in the COMMAND field and press Enter. This sets a breakpoint on the first program line.
7. To see where the breakpoints have been set, type **SHOW BREAKS** and press Enter. Figure 3-8 appears.

   This screen shows all breakpoints that have been set.

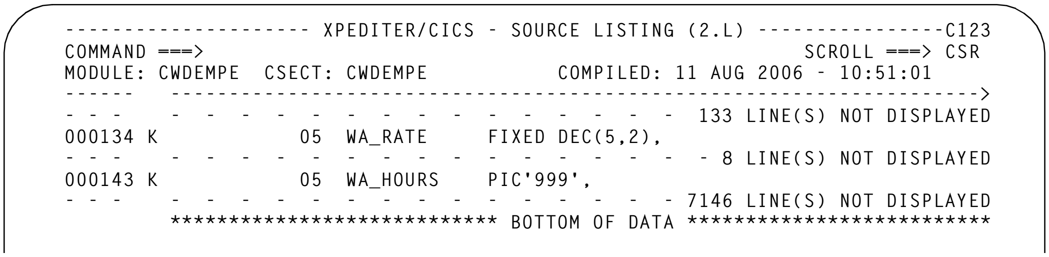
Figure 3-8. Source Listing Screen (2.L) Showing Breakpoints



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 11 AUG 2006 - 10:51:01
------   ------------------------------------------------------------------->
- - -    -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  360 LINE(S) NOT DISPLAYED
000361 B     CALL DFHENTRY_BF3CF5BF_B4ED7F82('   .  ................00000315' /
- - -    -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  480 LINE(S) NOT DISPLAYED
000842 B                   CURR_PAY   =  WA_HOURS * WA_RATE ;
-COND-   IF WA_HOURS EQ T"$$$"
- - -    -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 6447 LINE(S) NOT DISPLAYED
         **************************** BOTTOM OF DATA **************************
```

8. To see the keeps that have been set, type **SHOW KEEPS** and press Enter. See Figure 3-9.

Figure 3-9. Source Listing Screen (2.L) Showing Keeps



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 11 AUG 2006 - 10:51:01
------   ------------------------------------------------------------------->
- - -    -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  133 LINE(S) NOT DISPLAYED
000134 K             05  WA_RATE     FIXED DEC(5,2),
- - -    -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 8 LINE(S) NOT DISPLAYED
000143 K             05  WA_HOURS    PIC'999',
- - -    -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 7146 LINE(S) NOT DISPLAYED
         **************************** BOTTOM OF DATA **************************
```

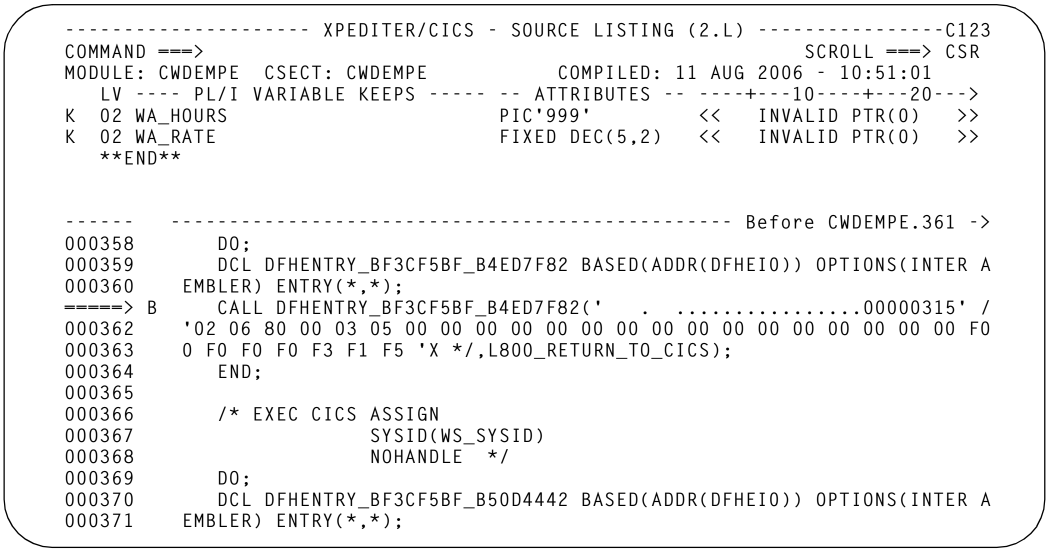
9. Type **RESET** and press Enter to display all source lines.

   In this example, you have set two breakpoints and selected two data fields to be kept. The program is now ready to test.

Executing the Program

1. Press Clear to return to CICS.
2. Type **XPLE** and press Enter. The first breakpoint in the program is taken and the Source Listing screen (2.L) appears (Figure 3-10).
3. Press PF12 (GO) to continue processing. The Demonstration Transaction screen appears.
4. Type **00001** and press Enter. The program stops again at the first statement.

Figure 3-10. Stopping at the First Statement on the Source Listing Screen (2.L)

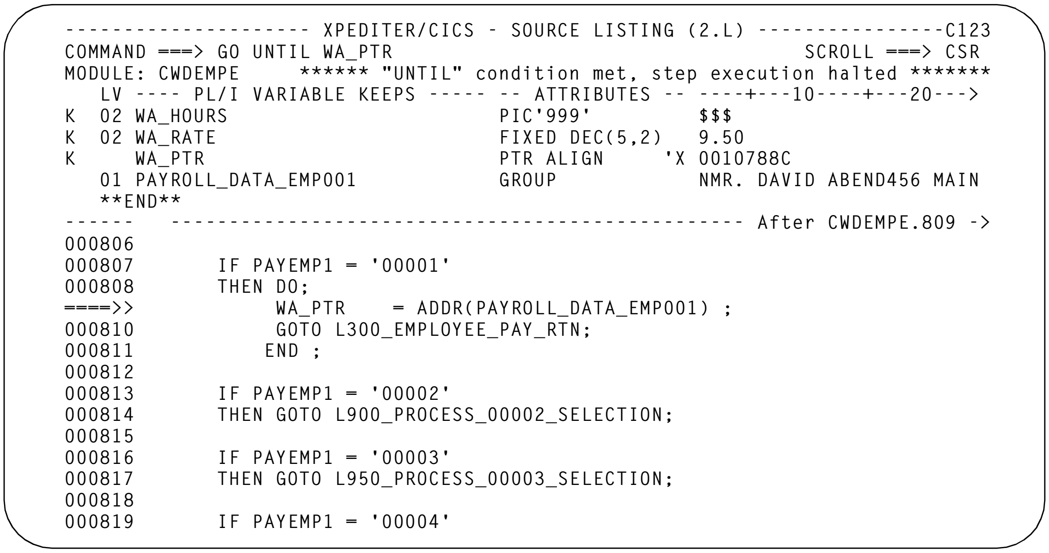


```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 11 AUG 2006 - 10:51:01
   LV ---- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
K  02 WA_HOURS                        PIC'999'         <<   INVALID PTR(0)   >>
K  02 WA_RATE                         FIXED DEC(5,2)   <<   INVALID PTR(0)   >>
   **END**


------   ------------------------------------------------ Before CWDEMPE.361 ->
000358       DO;
000359       DCL DFHENTRY_BF3CF5BF_B4ED7F82 BASED(ADDR(DFHEI0)) OPTIONS(INTER A
000360    EMBLER) ENTRY(*,*);
=====> B     CALL DFHENTRY_BF3CF5BF_B4ED7F82('   .  ................00000315' /
000362    '02 06 80 00 03 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0
000363    0 F0 F0 F0 F3 F1 F5 'X */,L800_RETURN_TO_CICS);
000364       END;
000365
000366       /* EXEC CICS ASSIGN
000367                    SYSID(WS_SYSID)
000368                    NOHANDLE  */
000369       DO;
000370       DCL DFHENTRY_BF3CF5BF_B50D4442 BASED(ADDR(DFHEI0)) OPTIONS(INTER A
000371    EMBLER) ENTRY(*,*);
```

Look at the value for WA_HOURS. The indicator << **INVALID PTR(0)** >> shows that WA_HOURS is part of a based variable that depends on a pointer that is not initialized.

5. Type **FIND WA_HOURS PREV** and press Enter to see where WA_HOURS is declared.
6. Scroll back to the declaration of the structure (WORK_AREA). Note that the structure is based on WA_PTR.
7. Type **GO UNTIL WA_PTR** in the COMMAND field and press Enter. This command tells Xpediter/CICS to execute the program until the value of WA_PTR changes (Figure 3-11).

Figure 3-11. Executing Until a Data Name Changes on the Source Listing Screen (2.L)



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===> GO UNTIL WA_PTR                                  SCROLL ===> CSR
MODULE: CWDEMPE     ****** "UNTIL" condition met, step execution halted *******
   LV ---- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
K  02 WA_HOURS                        PIC'999'         $$$
K  02 WA_RATE                         FIXED DEC(5,2)   9.50
K     WA_PTR                          PTR ALIGN     'X 0010788C
   01 PAYROLL_DATA_EMP001             GROUP            NMR. DAVID ABEND456 MAIN
   **END**
------   ------------------------------------------------- After CWDEMPE.809 ->
000806
000807       IF PAYEMP1 = '00001'
000808       THEN DO;
====>>            WA_PTR    = ADDR(PAYROLL_DATA_EMP001) ;
000810            GOTO L300_EMPLOYEE_PAY_RTN;
000811           END ;
000812
000813       IF PAYEMP1 = '00002'
000814       THEN GOTO L900_PROCESS_00002_SELECTION;
000815
000816       IF PAYEMP1 = '00003'
000817       THEN GOTO L950_PROCESS_00003_SELECTION;
000818
000819       IF PAYEMP1 = '00004'
```

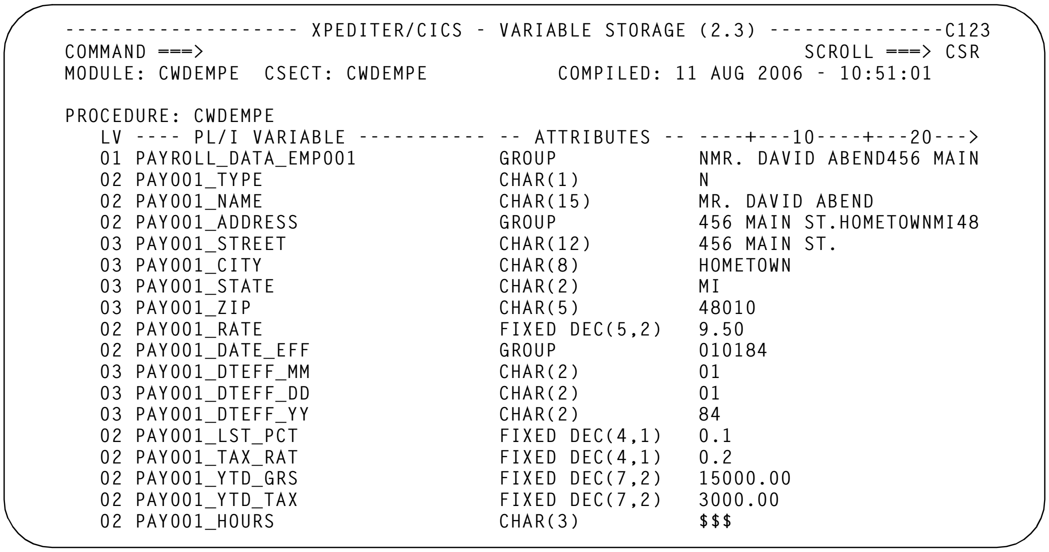
Look at the current statement. PAYROLL_DATA_EMP001 is a group item that contains the $$$ data. Next we need to obtain a more detailed look at this field.

8. Clear the COMMAND line by pressing EOF.

Note: The following step requires that CSR be entered in your SCROLL field.

9. Position the cursor in the keep window on PAYROLL_DATA_EMP001 and press PF8 (DOWN) to move this field to the top of the display.
10. Press PF16 (WS) to transfer to the Variable Storage screen (2.3). Note that you are positioned with PAYROLL_DATA_EMP001 at the top of the screen. The first item in the keep window always appears first on the Variable Storage screen (2.3) (Figure 3-12).

Figure 3-12. Variable Storage Screen (2.3)



```
------------------- XPEDITER/CICS - VARIABLE STORAGE (2.3) ---------------C123
COMMAND ===>                                                 SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 11 AUG 2006 - 10:51:01

PROCEDURE: CWDEMPE
   LV ---- PL/I VARIABLE ----------- -- ATTRIBUTES -- ----+---10----+---20--->
   01 PAYROLL_DATA_EMP001             GROUP            NMR. DAVID ABEND456 MAIN
   02 PAY001_TYPE                     CHAR(1)          N
   02 PAY001_NAME                     CHAR(15)         MR. DAVID ABEND
   02 PAY001_ADDRESS                  GROUP            456 MAIN ST.HOMETOWNMI48
   03 PAY001_STREET                   CHAR(12)         456 MAIN ST.
   03 PAY001_CITY                     CHAR(8)          HOMETOWN
   03 PAY001_STATE                    CHAR(2)          MI
   03 PAY001_ZIP                      CHAR(5)          48010
   02 PAY001_RATE                     FIXED DEC(5,2)   9.50
   02 PAY001_DATE_EFF                 GROUP            010184
   03 PAY001_DTEFF_MM                 CHAR(2)          01
   03 PAY001_DTEFF_DD                 CHAR(2)          01
   03 PAY001_DTEFF_YY                 CHAR(2)          84
   02 PAY001_LST_PCT                  FIXED DEC(4,1)   0.1
   02 PAY001_TAX_RAT                  FIXED DEC(4,1)   0.2
   02 PAY001_YTD_GRS                  FIXED DEC(7,2)   15000.00
   02 PAY001_YTD_TAX                  FIXED DEC(7,2)   3000.00
   02 PAY001_HOURS                    CHAR(3)          $$$
```

Look at the value of PAY001_HOURS at the bottom of the screen. This is where the $$$ originated.

11. Type the **K** line command to the left of PAY001_HOURS.
12. Correct the value by typing **040** over the $$$ and pressing Enter.

Resuming Execution at Another Statement

To correct the value in WA_HOURS and avoid an abend, re-execute the statement that originally moved bad data to WA_HOURS.

1. Press PF6 (LOCATE *) to return to the Source Listing screen (2.L).
2. Type **GT** on the IF PAYEMP1 = '00001' line and press Enter. This indicates that execution is to be resumed at this line.

   Notice that the values of the data field in the keep window change.

Stepping and Reviewing Program Execution

The GO command is used to resume program execution. In its simplest format, GO resumes at the current statement. When *number of statements* and *delay* parameters are added, GO executes the program in slow motion, pausing between each statement.

1. Type **GO 6 1** in the COMMAND field and press Enter. The following actions occur:

   – Six lines of code are executed with a one-second pause between each statement.

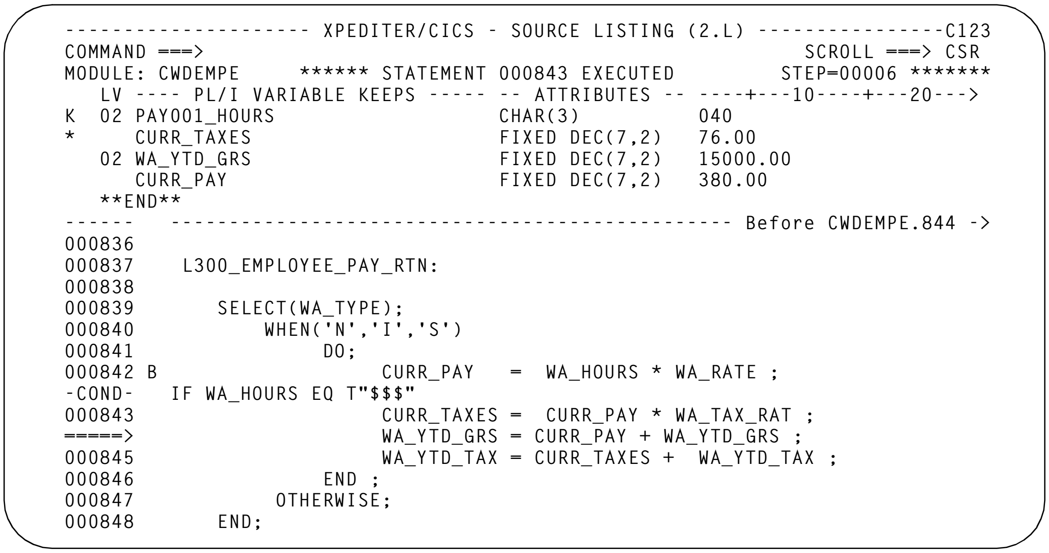
   – The execution pointer points to the current statement, which is highlighted.

   – The keep window displays the variable fields contained in the current statement and those you selected for viewing.

   – The screen shows the statements that have executed and how many steps are complete (Figure 3-13).

   The Program Trace screen (2.4) in Xpediter/CICS displays the instructions that have executed during your test and gives an instruction-by-instruction trace of the execution of your programs. You can use this trace to check the logic flow.

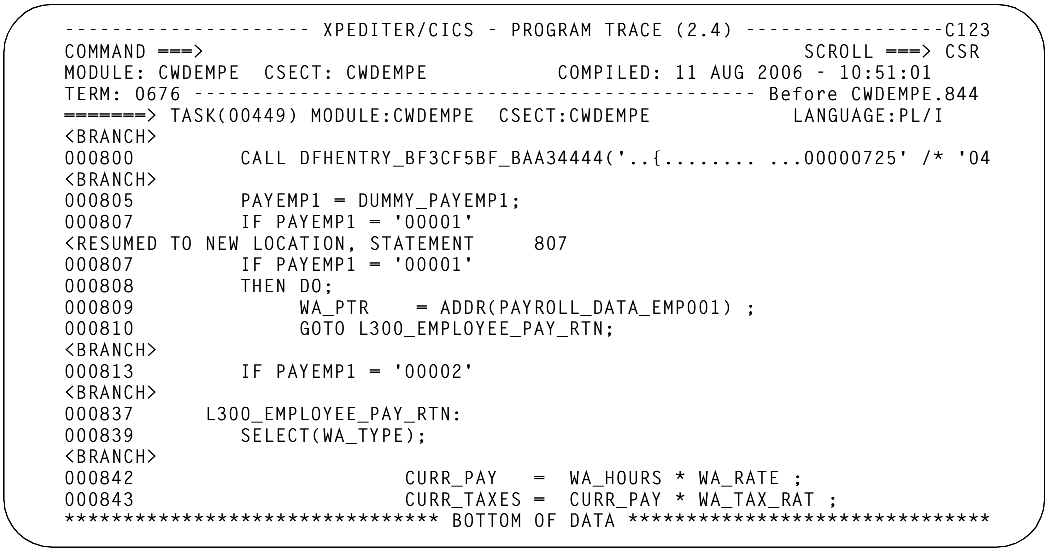
Figure 3-13. Stepping Through Code on the Source Listing Screen (2.L)



```
--------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                       SCROLL ===> CSR
MODULE: CWDEMPE     ****** STATEMENT 000843 EXECUTED         STEP=00006 *******
   LV ---- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
K  02 PAY001_HOURS                  CHAR(3)          040
*     CURR_TAXES                    FIXED DEC(7,2)   76.00
   02 WA_YTD_GRS                    FIXED DEC(7,2)   15000.00
      CURR_PAY                      FIXED DEC(7,2)   380.00
   **END**
------   ----------------------------------------------- Before CWDEMPE.844 ->
000836
000837    L300_EMPLOYEE_PAY_RTN:
000838
000839       SELECT(WA_TYPE);
000840           WHEN('N','I','S')
000841                DO;
000842 B                   CURR_PAY   =  WA_HOURS * WA_RATE ;
-COND-   IF WA_HOURS EQ T"$$$"
000843                     CURR_TAXES =  CURR_PAY * WA_TAX_RAT ;
=====>                     WA_YTD_GRS = CURR_PAY + WA_YTD_GRS ;
000845                     WA_YTD_TAX = CURR_TAXES +  WA_YTD_TAX ;
000846                END ;
000847            OTHERWISE;
000848       END;
```

2. Press PF17 (TRACE) to transfer to the Program Trace screen (2.4) (Figure 3-14). This screen shows the execution of your program logic with the current instruction positioned at the bottom of the screen. You can scroll through the data on this screen.

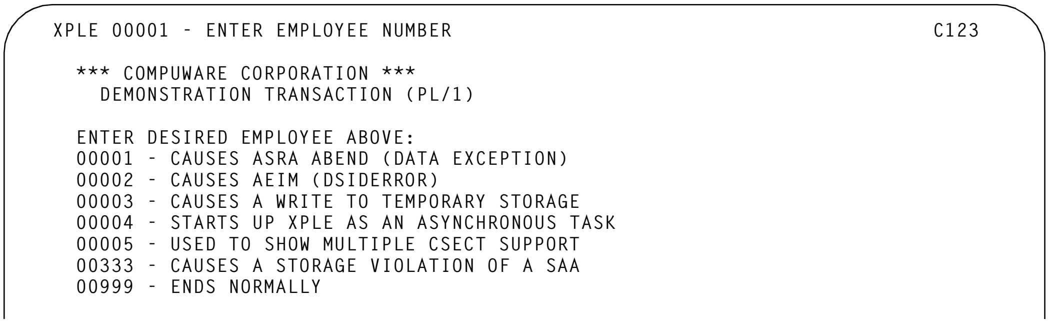
Figure 3-14. Program Trace Screen (2.4)



```
-------------------- XPEDITER/CICS - PROGRAM TRACE (2.4) -------------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 11 AUG 2006 - 10:51:01
TERM: 0676 ------------------------------------------------- Before CWDEMPE.844
=======> TASK(00449) MODULE:CWDEMPE  CSECT:CWDEMPE            LANGUAGE:PL/I
<BRANCH>
000800         CALL DFHENTRY_BF3CF5BF_BAA34444('..{........ ...00000725' /* '04
<BRANCH>
000805         PAYEMP1 = DUMMY_PAYEMP1;
000807         IF PAYEMP1 = '00001'
<RESUMED TO NEW LOCATION, STATEMENT     807
000807         IF PAYEMP1 = '00001'
000808         THEN DO;
000809              WA_PTR    = ADDR(PAYROLL_DATA_EMP001) ;
000810              GOTO L300_EMPLOYEE_PAY_RTN;
<BRANCH>
000813         IF PAYEMP1 = '00002'
<BRANCH>
000837      L300_EMPLOYEE_PAY_RTN:
000839         SELECT(WA_TYPE);
<BRANCH>
000842                       CURR_PAY   =  WA_HOURS * WA_RATE ;
000843                       CURR_TAXES =  CURR_PAY * WA_TAX_RAT ;
******************************** BOTTOM OF DATA ********************************
```

3. Press PF18 (Last 3270) to display the last 3270 screen (Figure 3-15). This allows you to verify what was entered on the screen that may have caused the application to take the particular logic path.

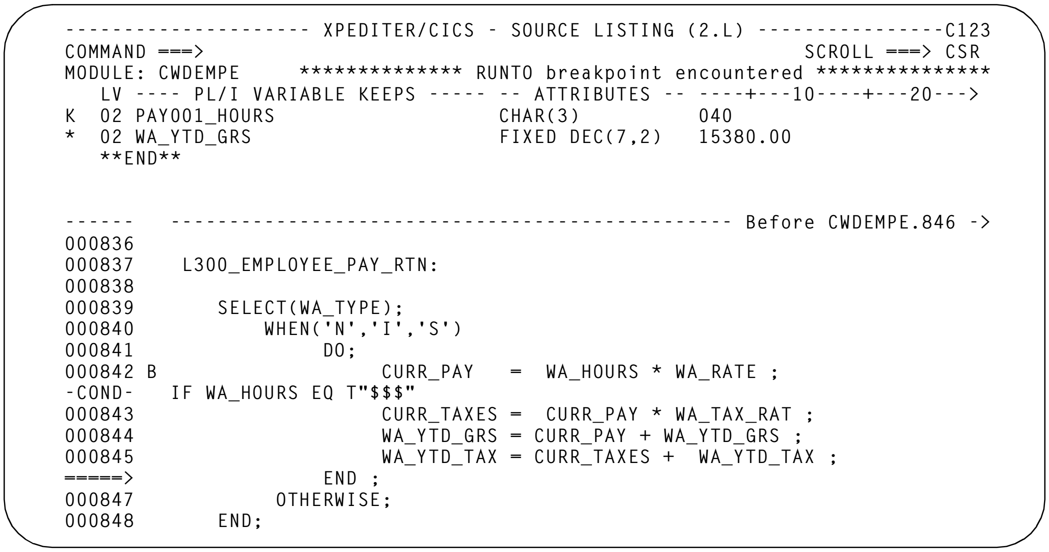
Figure 3-15. Last 3270 Screen



```
XPLE 00001 - ENTER EMPLOYEE NUMBER                                          C123

  *** COMPUWARE CORPORATION ***
    DEMONSTRATION TRANSACTION (PL/1)

  ENTER DESIRED EMPLOYEE ABOVE:
  00001 - CAUSES ASRA ABEND (DATA EXCEPTION)
  00002 - CAUSES AEIM (DSIDERROR)
  00003 - CAUSES A WRITE TO TEMPORARY STORAGE
  00004 - STARTS UP XPLE AS AN ASYNCHRONOUS TASK
  00005 - USED TO SHOW MULTIPLE CSECT SUPPORT
  00333 - CAUSES A STORAGE VIOLATION OF A SAA
  00999 - ENDS NORMALLY
```

4. Press Enter to return the Program Trace screen (2.4).
5. Press PF6 (LOCATE *) to return to the Source Listing screen (2.L) and position to the current statement.
6. Place the cursor on the first END statement below the current location, then press PF2 (RUNTO CSR). As shown in Figure 3-16 on page 3-11, execution halts on the END statement, where the cursor was positioned when PF2 was pressed. Pressing PF2 caused a runto breakpoint to be set on the statement where the cursor was positioned, a GO command to be issued, and the runto breakpoint to be deleted automatically after it was taken.

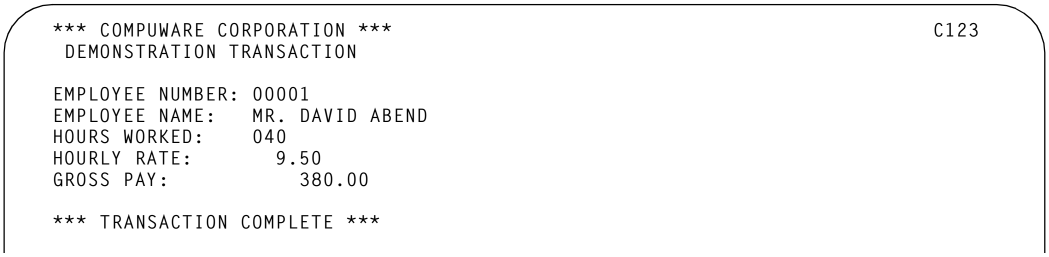
Figure 3-16. PF2 (RUNTO CSR) on the Source Listing Screen (2.L)



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ---------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE: CWDEMPE     ************** RUNTO breakpoint encountered ***************
   LV ---- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
K  02 PAY001_HOURS                   CHAR(3)          040
*  02 WA_YTD_GRS                     FIXED DEC(7,2)   15380.00
   **END**


------   ------------------------------------------------ Before CWDEMPE.846 ->
000836
000837    L300_EMPLOYEE_PAY_RTN:
000838
000839       SELECT(WA_TYPE);
000840           WHEN('N','I','S')
000841                DO;
000842 B                   CURR_PAY   =  WA_HOURS * WA_RATE ;
-COND-   IF WA_HOURS EQ T"$$$"
000843                     CURR_TAXES =  CURR_PAY * WA_TAX_RAT ;
000844                     WA_YTD_GRS = CURR_PAY + WA_YTD_GRS ;
000845                     WA_YTD_TAX = CURR_TAXES +  WA_YTD_TAX ;
=====>                  END ;
000847            OTHERWISE;
000848       END;
```

7. Press PF12 (GO) to resume execution (Figure 3-17).

Figure 3-17. Demonstration Transaction Screen



```
*** COMPUWARE CORPORATION ***                                        C123
 DEMONSTRATION TRANSACTION

EMPLOYEE NUMBER: 00001
EMPLOYEE NAME:   MR. DAVID ABEND
HOURS WORKED:    040
HOURLY RATE:       9.50
GROSS PAY:           380.00

*** TRANSACTION COMPLETE ***
```

Remember to end your session as described in “Exiting Xpediter/CICS and Ending a Debugging Session” on page 2-6.

Intercept Summary (1.7)

Before breakpoints can also be set using the Intercept Summary (1.7) screen. This provides an easy method for entering multiple intercept commands at one time. Before breakpoints set at the entry point of a CSECT are considered intercepts.

Note: Source listings must be available to set intercepts.

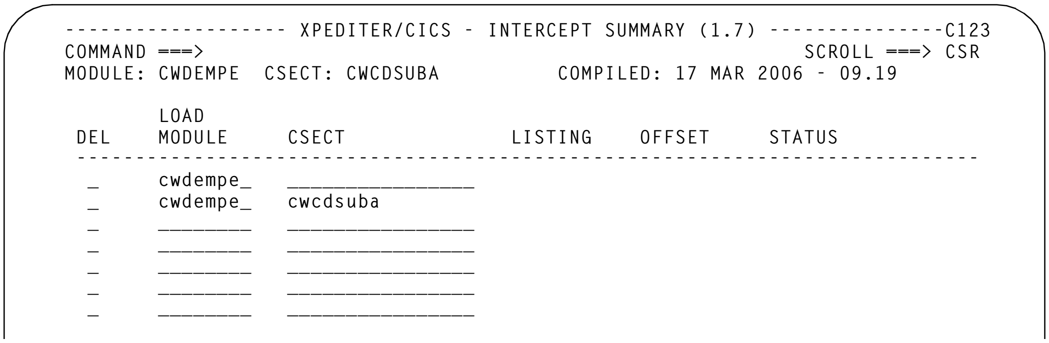
1. Access the Intercept Summary screen by entering 7 on the Session Control (1) Menu or by entering **=1.7** in the COMMAND field.

   If any current intercepts for your terminal exist at this time, they are displayed when the initial screen appears, followed by lines with underscores to enter new data. You can enter either the load module name or a load module and a CSECT name for intercepts you want to set. If only the load module name is entered, it is propagated into the CSECT name to set the intercept.

2. Enter **CWDEMPE** on the first line under LOAD MODULE.

3. Enter **CWDEMPE** on the second line under LOAD MODULE, then enter **CWCDSUBA** under CSECT.

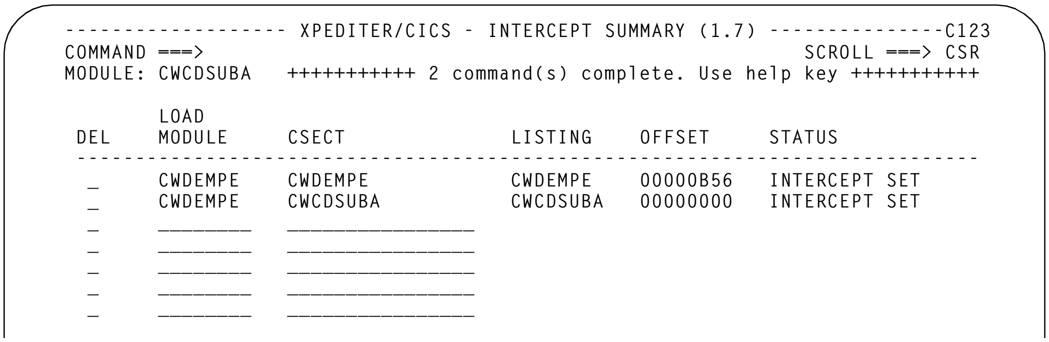
Figure 3-18. Intercept Summary (1.7)



```
------------------ XPEDITER/CICS - INTERCEPT SUMMARY (1.7) ---------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWCDSUBA          COMPILED: 17 MAR 2006 - 09.19

        LOAD
 DEL    MODULE     CSECT              LISTING    OFFSET     STATUS
 -----------------------------------------------------------------------------
  _     cwdempe_   ________________
  _     cwdempe_   cwcdsuba
  _     ________   ________________
  _     ________   ________________
  _     ________   ________________
  _     ________   ________________
  _     ________   ________________
```

4. Press Enter. The screen is refreshed and the result of each new line command is displayed. If the intercept was successfully set, the LISTING and OFFSET fields are updated and the STATUS column shows INTERCEPT SET.

Figure 3-19. Intercept Summary with Intercept Set

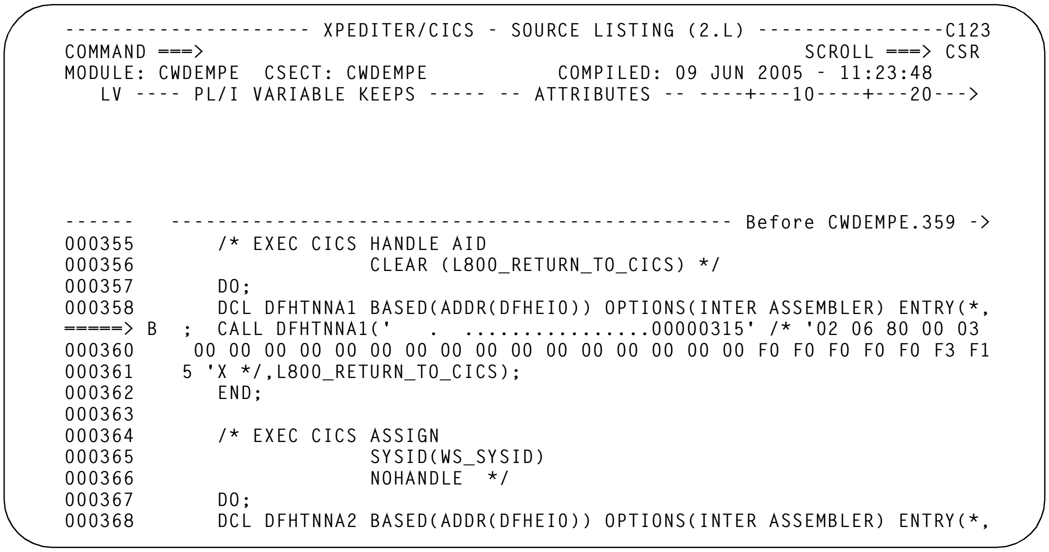


```
------------------ XPEDITER/CICS - INTERCEPT SUMMARY (1.7) ---------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE: CWCDSUBA   +++++++++++ 2 command(s) complete. Use help key +++++++++++

        LOAD
 DEL    MODULE     CSECT              LISTING    OFFSET     STATUS
 -----------------------------------------------------------------------------
  _     CWDEMPE    CWDEMPE            CWDEMPE    00000B56   INTERCEPT SET
  _     CWDEMPE    CWCDSUBA           CWCDSUBA   00000000   INTERCEPT SET
  _     ________   ________________
  _     ________   ________________
  _     ________   ________________
  _     ________   ________________
  _     ________   ________________
```

If there was an error, the data in error is highlighted and an error message is displayed under STATUS. In our example, both intercepts were successfully set.

5. Clear your screen.
6. Enter **XPLE**. A before breakpoint is taken at the entry point of CWDEMPE.

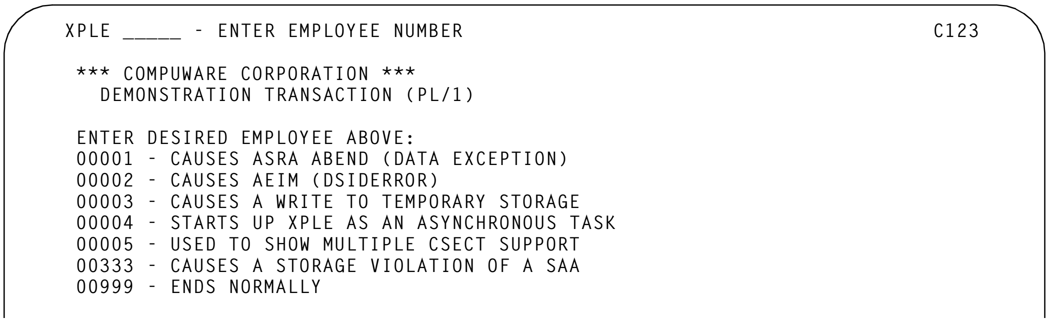
Figure 3-20. Before Breakpoint Taken



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ---------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
   LV ---- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->





------   ------------------------------------------------ Before CWDEMPE.359 ->
000355       /* EXEC CICS HANDLE AID
000356                    CLEAR (L800_RETURN_TO_CICS) */
000357       DO;
000358       DCL DFHTNNA1 BASED(ADDR(DFHEIO)) OPTIONS(INTER ASSEMBLER) ENTRY(*,
=====> B  ;  CALL DFHTNNA1('   .  ................00000315' /* '02 06 80 00 03
000360     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0 F0 F0 F0 F0 F3 F1
000361    5 'X */,L800_RETURN_TO_CICS);
000362       END;
000363
000364       /* EXEC CICS ASSIGN
000365                    SYSID(WS_SYSID)
000366                    NOHANDLE  */
000367       DO;
000368       DCL DFHTNNA2 BASED(ADDR(DFHEIO)) OPTIONS(INTER ASSEMBLER) ENTRY(*,
```

7. After taking the first intercept at the entry point of CWDEMPE, press PF12 to continue the program and display the Demonstration Transaction screen requesting the employee number.

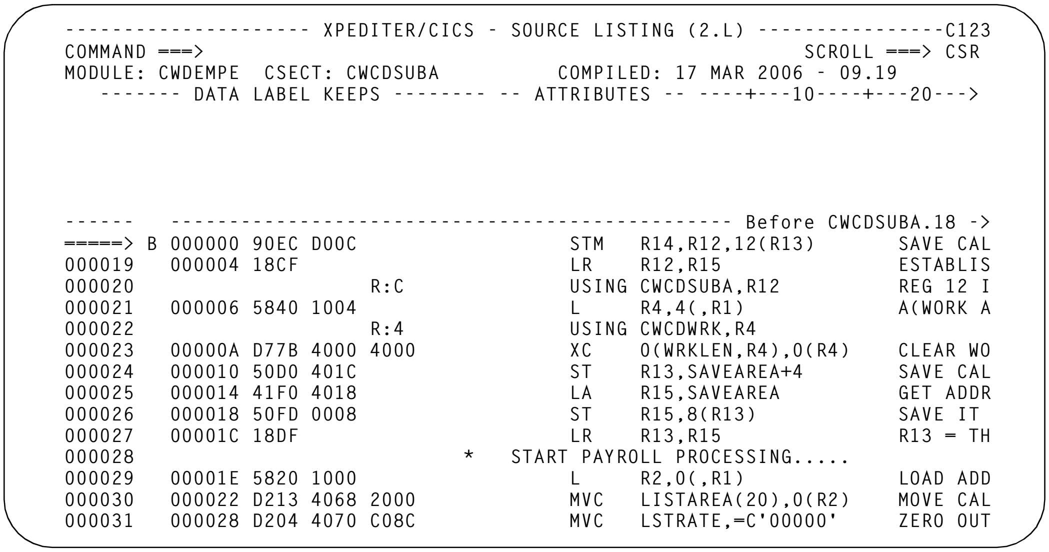
Figure 3-21. Demonstration Transaction Screen



```
XPLE _____ - ENTER EMPLOYEE NUMBER                                         C123

 *** COMPUWARE CORPORATION ***
   DEMONSTRATION TRANSACTION (PL/1)

 ENTER DESIRED EMPLOYEE ABOVE:
 00001 - CAUSES ASRA ABEND (DATA EXCEPTION)
 00002 - CAUSES AEIM (DSIDERROR)
 00003 - CAUSES A WRITE TO TEMPORARY STORAGE
 00004 - STARTS UP XPLE AS AN ASYNCHRONOUS TASK
 00005 - USED TO SHOW MULTIPLE CSECT SUPPORT
 00333 - CAUSES A STORAGE VIOLATION OF A SAA
 00999 - ENDS NORMALLY
```

8. Enter **00005** and press Enter. The initial intercept in CWDEMPE is taken again.
9. Press PF12 to continue. The second intercept in CWCDSUBA is taken.

Figure 3-22. Second Intercept in CWCDSUBA



```
--------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWCDSUBA          COMPILED: 17 MAR 2006 - 09.19
   ------- DATA LABEL KEEPS -------- -- ATTRIBUTES -- ----+---10----+---20--->





------   ------------------------------------------------ Before CWCDSUBA.18 ->
=====> B 000000 90EC D00C                  STM   R14,R12,12(R13)       SAVE CAL
000019   000004 18CF                       LR    R12,R15               ESTABLIS
000020                    R:C              USING CWCDSUBA,R12          REG 12 I
000021   000006 5840 1004                  L     R4,4(,R1)             A(WORK A
000022                    R:4              USING CWCDWRK,R4
000023   00000A D77B 4000 4000             XC    0(WRKLEN,R4),0(R4)    CLEAR WO
000024   000010 50D0 401C                  ST    R13,SAVEAREA+4        SAVE CAL
000025   000014 41F0 4018                  LA    R15,SAVEAREA          GET ADDR
000026   000018 50FD 0008                  ST    R15,8(R13)            SAVE IT
000027   00001C 18DF                       LR    R13,R15               R13 = TH
000028                            *   START PAYROLL PROCESSING.....
000029   00001E 5820 1000                  L     R2,0(,R1)             LOAD ADD
000030   000022 D213 4068 2000             MVC   LISTAREA(20),0(R2)    MOVE CAL
000031   000028 D204 4070 C08C             MVC   LSTRATE,=C'00000'     ZERO OUT
```

10. You can now end your debugging session as described in "Exiting Xpediter/CICS and Ending a Debugging Session" on page 2-6.

Using Enhanced Traps

"Setting Program Breakpoints" on page 3-5 gave an example of how to set a breakpoint to halt execution. In this section, we will create an enhanced trap and discuss its affect on breakpoints and abends.

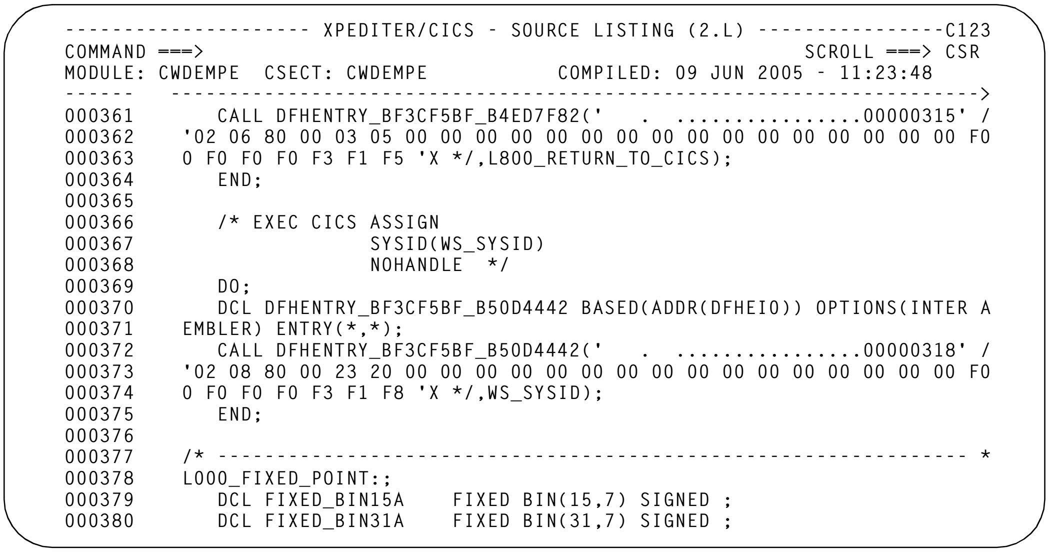
Enhanced trapping is an extension of regular trapping. A trap can be enhanced so breakpoints and abends are only taken when the specified condition is met. That condition can be a value in the initial COMMAREA (ICA), in a specific "Big Commarea" container (CONT+*containername*), MQ message descriptor (MQMD), or MQ message data (MQD). For more information, see the description of the Trap Summary screen (1.6 or 9.6) in the *Xpediter/CICS Reference Manual*.

Setting an Enhanced Trap on Initial COMMAREA (ICA)

Note: If you are using channels and containers instead of COMMAREAS, skip to "Setting an Enhanced Trap Using Containers (CONT)" on page 3-17.

1. Type **XPED CWDEMPE** on a blank CICS screen and press Enter. Program CWDEMPE is displayed on the Source Listing screen (2.L) as shown in Figure 3-23.

Figure 3-23. CWDEMPE on the Source Listing Screen (2.L)

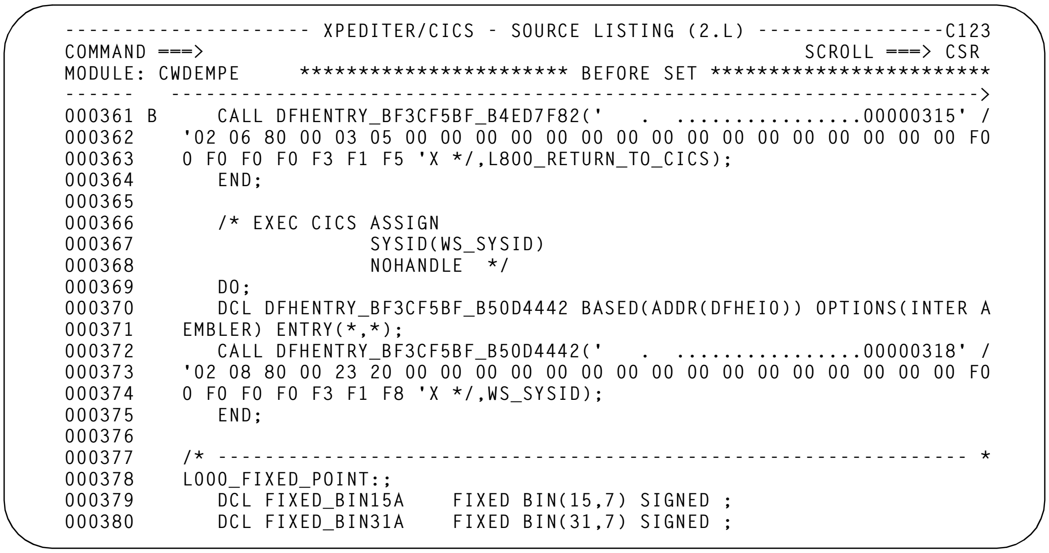


```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ---------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48
------   ------------------------------------------------------------------->
000361       CALL DFHENTRY_BF3CF5BF_B4ED7F82('   .  ...............00000315' /
000362    '02 06 80 00 03 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0
000363    0 F0 F0 F0 F3 F1 F5 'X */,L800_RETURN_TO_CICS);
000364       END;
000365
000366       /* EXEC CICS ASSIGN
000367                    SYSID(WS_SYSID)
000368                    NOHANDLE  */
000369       DO;
000370       DCL DFHENTRY_BF3CF5BF_B50D4442 BASED(ADDR(DFHEI0)) OPTIONS(INTER A
000371    EMBLER) ENTRY(*,*);
000372       CALL DFHENTRY_BF3CF5BF_B50D4442('   .  ...............00000318' /
000373    '02 08 80 00 23 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0
000374    0 F0 F0 F0 F3 F1 F8 'X */,WS_SYSID);
000375       END;
000376
000377    /* ------------------------------------------------------------- *
000378    L000_FIXED_POINT:;
000379       DCL FIXED_BIN15A    FIXED BIN(15,7) SIGNED ;
000380       DCL FIXED_BIN31A    FIXED BIN(31,7) SIGNED ;
```

2. Type **BEFORE 0** in the COMMAND field and press Enter. This sets a before breakpoint on the first executable statement in CWDEMPE. As shown in Figure 3-24, Xpediter displays the message

```
*********************** BEFORE SET ***********************
```

Figure 3-24. Setting a Breakpoint on the Source Listing Screen (2.L)



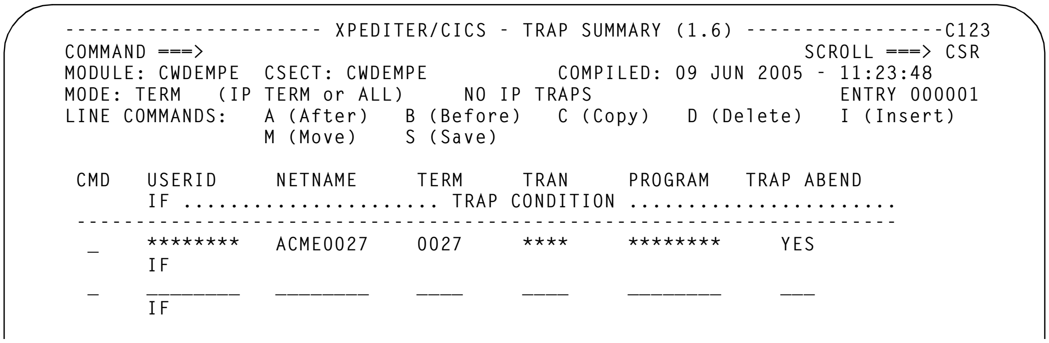
```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ---------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE: CWDEMPE     *********************** BEFORE SET ***********************
------   ------------------------------------------------------------------->
000361 B     CALL DFHENTRY_BF3CF5BF_B4ED7F82('   .  ...............00000315' /
000362    '02 06 80 00 03 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0
000363    0 F0 F0 F0 F3 F1 F5 'X */,L800_RETURN_TO_CICS);
000364       END;
000365
000366       /* EXEC CICS ASSIGN
000367                    SYSID(WS_SYSID)
000368                    NOHANDLE  */
000369       DO;
000370       DCL DFHENTRY_BF3CF5BF_B50D4442 BASED(ADDR(DFHEI0)) OPTIONS(INTER A
000371    EMBLER) ENTRY(*,*);
000372       CALL DFHENTRY_BF3CF5BF_B50D4442('   .  ...............00000318' /
000373    '02 08 80 00 23 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0
000374    0 F0 F0 F0 F3 F1 F8 'X */,WS_SYSID);
000375       END;
000376
000377    /* ------------------------------------------------------------- *
000378    L000_FIXED_POINT:;
000379       DCL FIXED_BIN15A    FIXED BIN(15,7) SIGNED ;
000380       DCL FIXED_BIN31A    FIXED BIN(31,7) SIGNED ;
```

Note: If your profile specifies TRAP=ON, starting Xpediter automatically creates a trap based on netname and terminal ID. If TRAP=OFF, the trap is created when you set a breakpoint.

3. Type **=1.6** in the COMMAND field and press Enter. The Trap Summary screen (1.6) is displayed (Figure 3-25) showing the trap Xpediter automatically created based on netname and terminal ID.

The breakpoints you set will only be taken when the task is running on netname ACME0027 and terminal 0027. Abends for that netname and terminal will also be trapped.

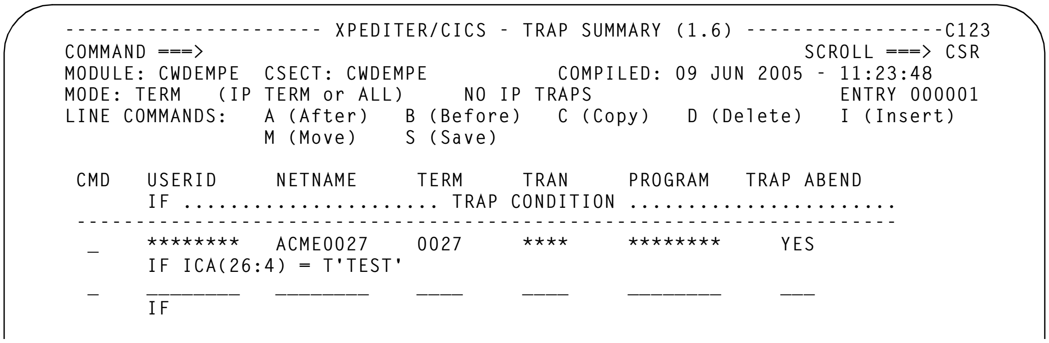
Figure 3-25. Displaying a Trap on the Trap Summary Screen (1.6)



```
---------------------- XPEDITER/CICS - TRAP SUMMARY (1.6) -----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48
MODE: TERM   (IP TERM or ALL)     NO IP TRAPS                     ENTRY 000001
LINE COMMANDS:   A (After)   B (Before)   C (Copy)   D (Delete)   I (Insert)
                 M (Move)    S (Save)

 CMD   USERID     NETNAME     TERM     TRAN     PROGRAM   TRAP ABEND
       IF ...................... TRAP CONDITION .......................
 ----------------------------------------------------------------------
  _    ********   ACME0027    0027     ****     ********     YES
       IF
  _    ________   ________    ____     ____     ________     ___
       IF
```

4. To create an enhanced trap, type **ICA(26:4)=T'TEST**' in the trap condition field on the second line of the trap entry following the word **IF** and press Enter.

Figure 3-26. Enhanced Trap for Initial COMMAREA



```
---------------------- XPEDITER/CICS - TRAP SUMMARY (1.6) -----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48
MODE: TERM   (IP TERM or ALL)     NO IP TRAPS                     ENTRY 000001
LINE COMMANDS:   A (After)   B (Before)   C (Copy)   D (Delete)   I (Insert)
                 M (Move)    S (Save)

 CMD   USERID     NETNAME     TERM     TRAN     PROGRAM   TRAP ABEND
       IF ...................... TRAP CONDITION .......................
 ----------------------------------------------------------------------
  _    ********   ACME0027    0027     ****     ********     YES
       IF ICA(26:4) = T'TEST'
  _    ________   ________    ____     ____     ________     ___
       IF
```

This enhanced trap causes Xpediter to take breakpoints and trap abends only when both of the following conditions are met:

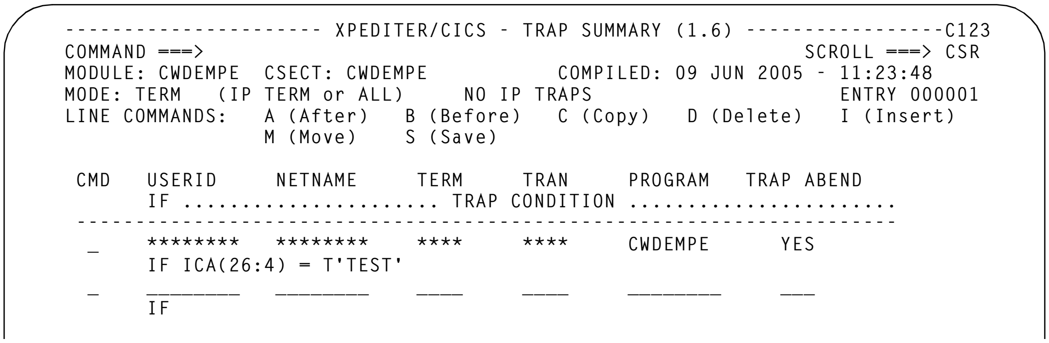
- The task is running on netname ACME0027 and terminal 0027, and
- The initial DFHCOMMAREA has a value of **TEST** in the four characters starting at position 26.

The literal **T'TEST**' could also have been entered as '**TEST**' without the preceding type specification of **T**. Because this text type literal is not case-sensitive, you could also have entered **T'test**', '**Test**', or '**TeSt**'.

5. The enhanced trap could also be modified to eliminate the netname and terminal ID requirements. As shown in Figure 3-27 on page 3-17, overtype **ACME0027** in the **NETNAME** field and **0027** in the **TERM** field with all asterisks (*****), type **CWDEMPE** in the **PROGRAM** field, and press Enter.

   This type of enhanced trap is useful if there are hundreds of terminals executing a single program, but you only want to stop in that program when the initial COMMAREA contains the specified value.

Figure 3-27. Enhanced Trap for All Netnames and Terminal IDs Running CWDEMPE

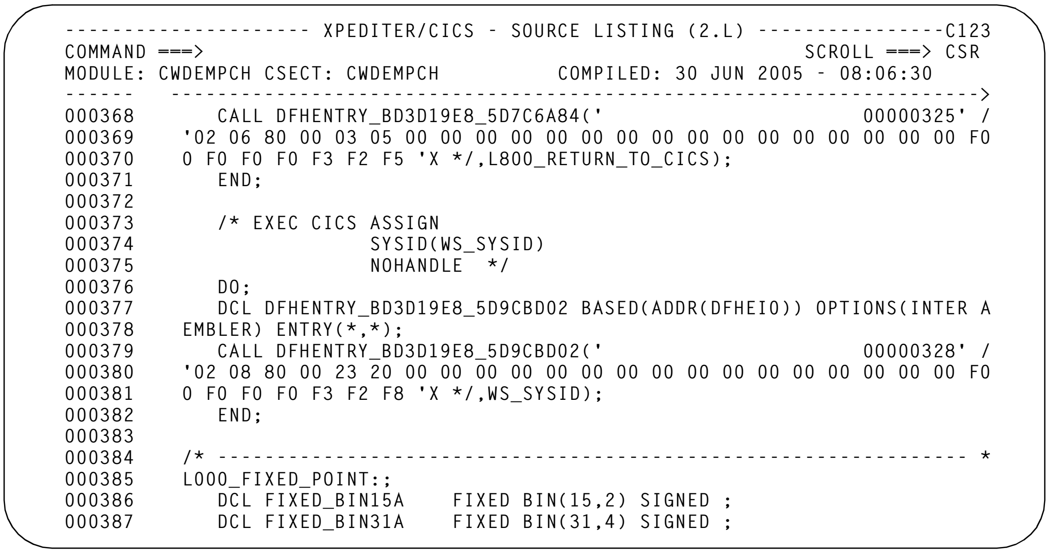


```
---------------------- XPEDITER/CICS - TRAP SUMMARY (1.6) -----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48
MODE: TERM   (IP TERM or ALL)     NO IP TRAPS                      ENTRY 000001
LINE COMMANDS:    A (After)   B (Before)   C (Copy)   D (Delete)   I (Insert)
                  M (Move)    S (Save)

 CMD   USERID     NETNAME     TERM     TRAN     PROGRAM   TRAP ABEND
       IF ...................... TRAP CONDITION .......................
 ----------------------------------------------------------------------
  _    ********   ********    ****     ****     CWDEMPE      YES
       IF ICA(26:4) = T'TEST'

  _    ________   ________    ____     ____     ________     ___
       IF
```

Setting an Enhanced Trap Using Containers (CONT)

1. Type **XPED CWDEMPCH** on a blank CICS screen and press Enter. Program CWDEMPCH is displayed on the Source Listing screen (2.L) as shown in Figure 3-28.

Figure 3-28. CWDEMPCH on the Source Listing Screen (2.L)

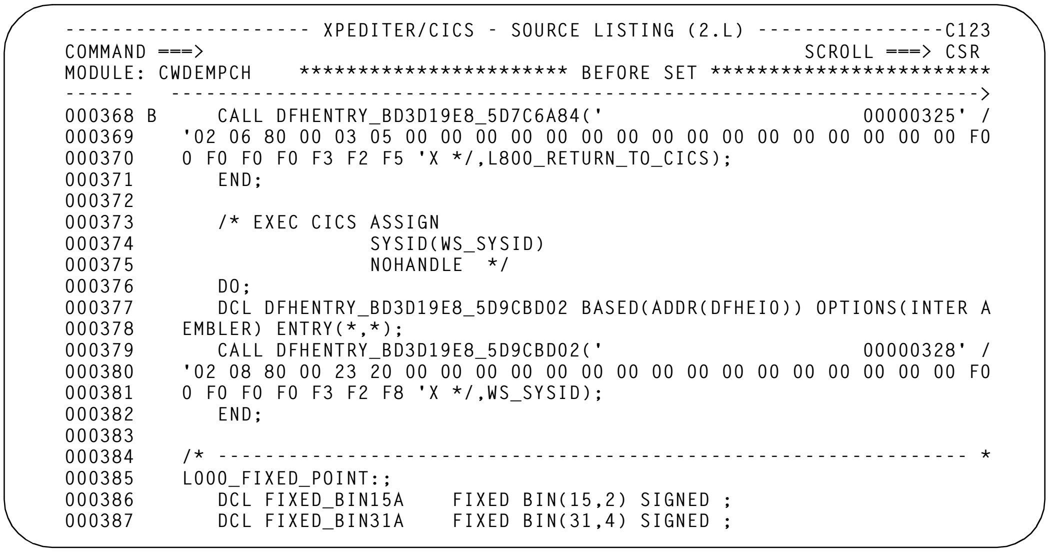


```
--------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPCH CSECT: CWDEMPCH         COMPILED: 30 JUN 2005 - 08:06:30
------   ------------------------------------------------------------------->
000368       CALL DFHENTRY_BD3D19E8_5D7C6A84('                      00000325' /
000369    '02 06 80 00 03 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0
000370    0 F0 F0 F0 F3 F2 F5 'X */,L800_RETURN_TO_CICS);
000371       END;
000372
000373       /* EXEC CICS ASSIGN
000374                    SYSID(WS_SYSID)
000375                    NOHANDLE  */
000376       DO;
000377       DCL DFHENTRY_BD3D19E8_5D9CBD02 BASED(ADDR(DFHEI0)) OPTIONS(INTER A
000378    EMBLER) ENTRY(*,*);
000379       CALL DFHENTRY_BD3D19E8_5D9CBD02('                      00000328' /
000380    '02 08 80 00 23 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0
000381    0 F0 F0 F0 F3 F2 F8 'X */,WS_SYSID);
000382       END;
000383
000384    /* ------------------------------------------------------------------ *
000385    L000_FIXED_POINT:;
000386       DCL FIXED_BIN15A    FIXED BIN(15,2) SIGNED ;
000387       DCL FIXED_BIN31A    FIXED BIN(31,4) SIGNED ;
```

2. Type **BEFORE 0** in the COMMAND field and press Enter. This sets a before breakpoint on the first executable statement in CWDEMPCH. As shown in Figure 3-29, Xpediter displays the message

```
********************** BEFORE SET ***********************
```

Figure 3-29. Setting a Breakpoint on the Source Listing Screen (2.L)



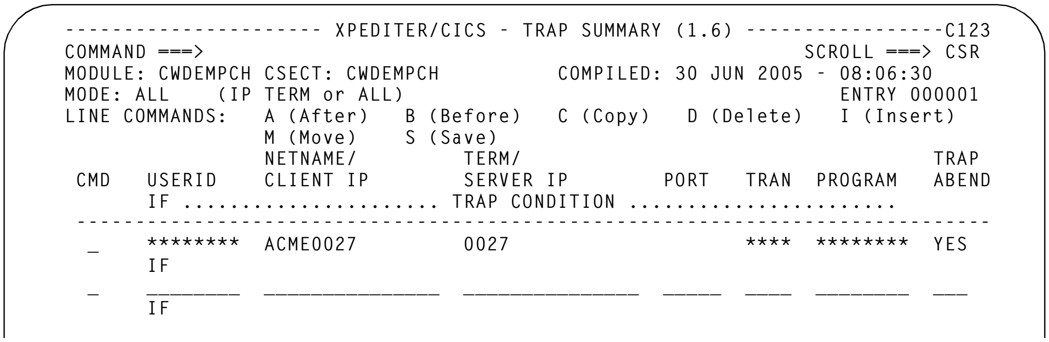
```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE: CWDEMPCH    *********************** BEFORE SET ***********************
------   -------------------------------------------------------------------->
000368 B     CALL DFHENTRY_BD3D19E8_5D7C6A84('                        00000325' /
000369    '02 06 80 00 03 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0
000370    0 F0 F0 F0 F3 F2 F5 'X */,L800_RETURN_TO_CICS);
000371       END;
000372
000373       /* EXEC CICS ASSIGN
000374                   SYSID(WS_SYSID)
000375                   NOHANDLE  */
000376       DO;
000377       DCL DFHENTRY_BD3D19E8_5D9CBD02 BASED(ADDR(DFHEIO)) OPTIONS(INTER A
000378    EMBLER) ENTRY(*,*);
000379       CALL DFHENTRY_BD3D19E8_5D9CBD02('                        00000328' /
000380    '02 08 80 00 23 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0
000381    0 F0 F0 F0 F3 F2 F8 'X */,WS_SYSID);
000382       END;
000383
000384    /* ------------------------------------------------------------------ *
000385    L000_FIXED_POINT:;
000386       DCL FIXED_BIN15A    FIXED BIN(15,2) SIGNED ;
000387       DCL FIXED_BIN31A    FIXED BIN(31,4) SIGNED ;
```

Note: If your profile specifies TRAP=ON, starting Xpediter automatically creates a trap based on netname and terminal ID. If TRAP=OFF, the trap is created when you set a breakpoint.

3. Type **=1.6** in the COMMAND field and press Enter. The Trap Summary screen (1.6) is displayed (Figure 3-30) showing the trap Xpediter automatically created based on netname and terminal ID.

   The breakpoints you set will only be taken when the task is running on netname ACME0027 and terminal 0027. Abends for that netname and terminal will also be trapped.

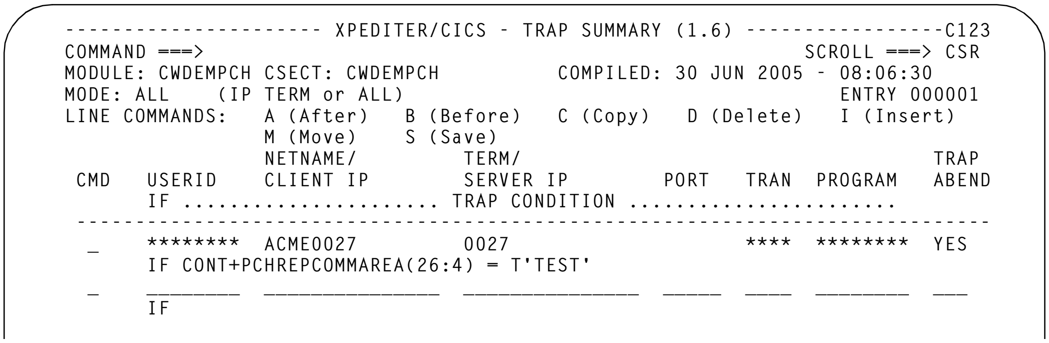
Figure 3-30. Displaying a Trap on the Trap Summary Screen (1.6)



```
--------------------- XPEDITER/CICS - TRAP SUMMARY (1.6) -----------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE: CWDEMPCH CSECT: CWDEMPCH          COMPILED: 30 JUN 2005 - 08:06:30
MODE: ALL    (IP TERM or ALL)                                   ENTRY 000001
LINE COMMANDS:   A (After)   B (Before)   C (Copy)   D (Delete)   I (Insert)
                 M (Move)    S (Save)
                 NETNAME/         TERM/                                   TRAP
 CMD   USERID    CLIENT IP        SERVER IP        PORT   TRAN  PROGRAM   ABEND
       IF ...................... TRAP CONDITION .......................
 -------------------------------------------------------------------------------
  _    ********  ACME0027         0027                    ****  ********  YES
       IF
  _    ________  _______________  _______________  _____  ____  ________  ___
       IF
```

4. To create an enhanced trap, type **CONT+PCHREPCOMMAREA(26:4)=T'TEST'** in the trap condition field on the second line of the trap entry following the word **IF** and press Enter.

Figure 3-31. Enhanced Trap for Container PCHREPCOMMAREA



```
--------------------- XPEDITER/CICS - TRAP SUMMARY (1.6) ----------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE: CWDEMPCH CSECT: CWDEMPCH         COMPILED: 30 JUN 2005 - 08:06:30
MODE: ALL    (IP TERM or ALL)                                     ENTRY 000001
LINE COMMANDS:   A (After)   B (Before)   C (Copy)   D (Delete)   I (Insert)
                 M (Move)    S (Save)
                 NETNAME/         TERM/                                   TRAP
 CMD   USERID    CLIENT IP        SERVER IP        PORT   TRAN  PROGRAM   ABEND
       IF ...................... TRAP CONDITION .......................
 -----------------------------------------------------------------------------
  _    ********  ACME0027         0027                    ****  ********  YES
       IF CONT+PCHREPCOMMAREA(26:4) = T'TEST'

  _    ________  _______________  _______________  _____  ____  ________  ___
       IF
```

This enhanced trap causes Xpediter to take breakpoints and trap abends only when both of the following conditions are met:

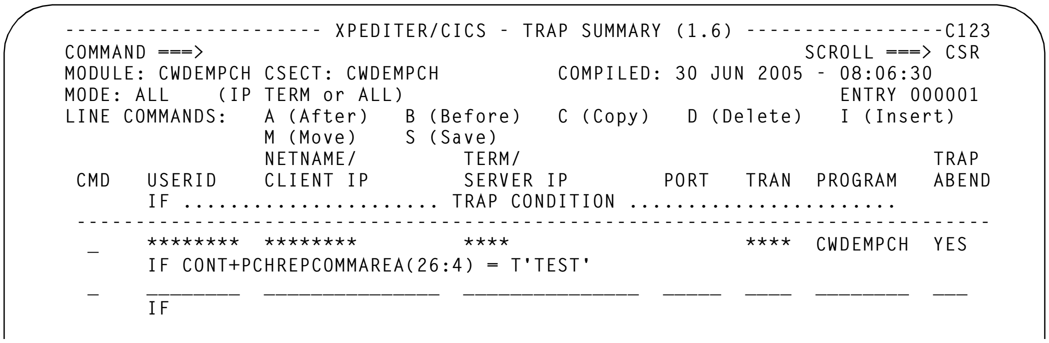
- The task is running on netname ACME0027 and terminal 0027, and
- The program has access to container PCHREPCOMMAREA and it has a value of **TEST** in the four characters starting at position 26.

The literal **T**'**TEST**' could also have been entered as '**TEST**' without the preceding type specification of **T**. Because this text type literal is not case-sensitive, you could also have entered **T**'**test**', '**Test**', or '**TeSt**'.

5. The enhanced trap could also be modified to eliminate the netname and terminal ID requirements. As shown in Figure 3-32, overtype **ACME0027** in the **NETNAME** field and **0027** in the **TERM** field with all asterisks (*****), type **CWDEMPCH** in the **PROGRAM** field, and press Enter.

   This type of enhanced trap is useful if there are hundreds of terminals executing a single program, but you only want to stop in that program when the named program's container contains the specified value.

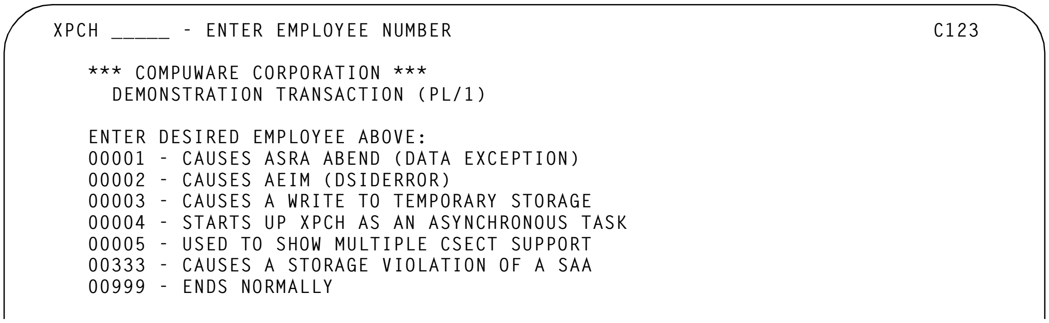
Figure 3-32. Enhanced Trap for All Netnames and Terminal IDs Running CWDEMPCH



```
--------------------- XPEDITER/CICS - TRAP SUMMARY (1.6) ----------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE: CWDEMPCH CSECT: CWDEMPCH         COMPILED: 30 JUN 2005 - 08:06:30
MODE: ALL    (IP TERM or ALL)                                     ENTRY 000001
LINE COMMANDS:   A (After)   B (Before)   C (Copy)   D (Delete)   I (Insert)
                 M (Move)    S (Save)
                 NETNAME/         TERM/                                   TRAP
 CMD   USERID    CLIENT IP        SERVER IP        PORT   TRAN  PROGRAM   ABEND
       IF ...................... TRAP CONDITION .......................
 -----------------------------------------------------------------------------
  _    ********  ********         ****                    ****  CWDEMPCH  YES
       IF CONT+PCHREPCOMMAREA(26:4) = T'TEST'

  _    ________  _______________  _______________  _____  ____  ________  ___
       IF
```

6. To see how this enhanced trap functions, first press Clear.
7. On the blank CICS screen, type **XPCH** and press Enter. The Demonstration Transaction screen is displayed as shown in Figure 3-33.

Figure 3-33. Demonstration Transaction Screen

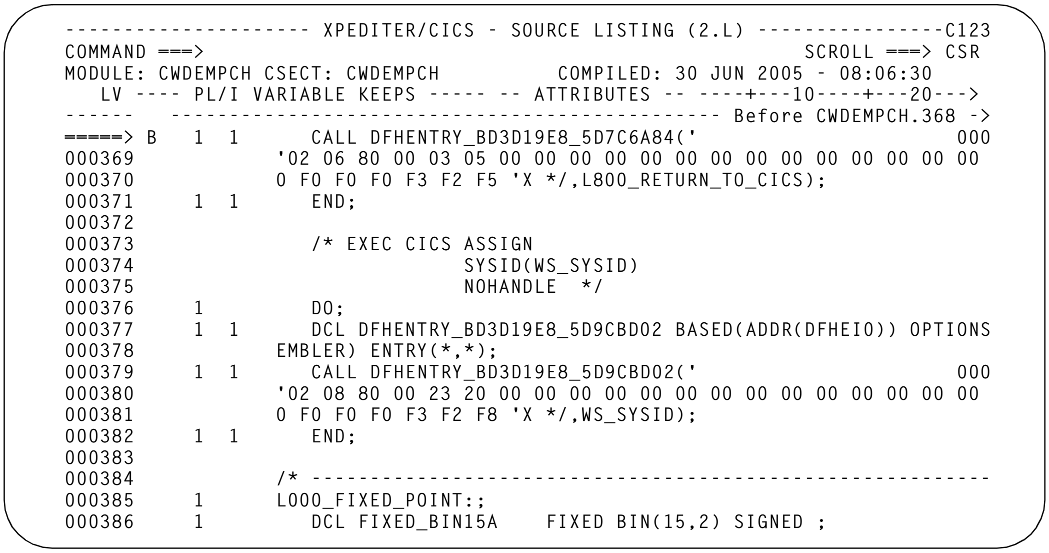


```
 XPCH _____ - ENTER EMPLOYEE NUMBER                                         C123

    *** COMPUWARE CORPORATION ***
      DEMONSTRATION TRANSACTION (PL/1)

    ENTER DESIRED EMPLOYEE ABOVE:
    00001 - CAUSES ASRA ABEND (DATA EXCEPTION)
    00002 - CAUSES AEIM (DSIDERROR)
    00003 - CAUSES A WRITE TO TEMPORARY STORAGE
    00004 - STARTS UP XPCH AS AN ASYNCHRONOUS TASK
    00005 - USED TO SHOW MULTIPLE CSECT SUPPORT
    00333 - CAUSES A STORAGE VIOLATION OF A SAA
    00999 - ENDS NORMALLY
```

Notice that the trap was not taken by Xpediter/CICS. This is because the initial invocation of the pseudo-conversational transaction XPCH does not have a container named PCHREPCOMMAREA with data for the trap criteria to match.

8. Type **00999** for the employee number and press Enter. Now Xpediter traps the transaction as shown in Figure 3-34. This is because the second invocation of the transaction was passed a container named PCHREPCOMMAREA with the characters "TEST" in positions 26 through 29 (26:4).

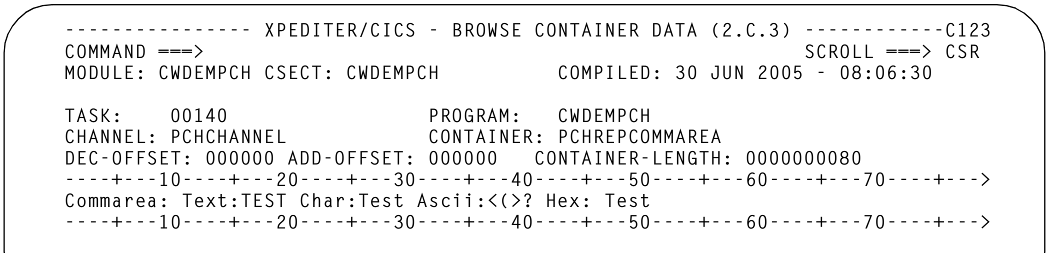
Figure 3-34. Taking an Enhanced Trap for Container PCHREPCOMMAREA



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ---------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE: CWDEMPCH CSECT: CWDEMPCH         COMPILED: 30 JUN 2005 - 08:06:30
   LV ---- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
------   -------------------------------------------- Before CWDEMPCH.368 ->
=====> B   1  1      CALL DFHENTRY_BD3D19E8_5D7C6A84('                      000
000369            '02 06 80 00 03 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000370            0 F0 F0 F0 F3 F2 F5 'X */,L800_RETURN_TO_CICS);
000371     1  1      END;
000372
000373               /* EXEC CICS ASSIGN
000374                           SYSID(WS_SYSID)
000375                           NOHANDLE  */
000376     1         DO;
000377     1  1      DCL DFHENTRY_BD3D19E8_5D9CBD02 BASED(ADDR(DFHEI0)) OPTIONS
000378            EMBLER) ENTRY(*,*);
000379     1  1      CALL DFHENTRY_BD3D19E8_5D9CBD02('                      000
000380            '02 08 80 00 23 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000381            0 F0 F0 F0 F3 F2 F8 'X */,WS_SYSID);
000382     1  1      END;
000383
000384            /* ----------------------------------------------------------
000385     1      L000_FIXED_POINT:;
000386     1         DCL FIXED_BIN15A    FIXED BIN(15,2) SIGNED ;
```

9. To confirm that the data in container PCHREPCOMMAREA satisfied your enhanced trap criteria, transfer to the Browse Container Data screen by typing =2.C.3 in the COMMAND field and pressing Enter. The current input channel and container appear as the defaults and TEST is displayed starting in position 26 of the container data in container PCHREPCOMMAREA as shown in Figure 3-35. This data is displayed in this sample in HEX OFF mode to show character format only. If you want to display another container's data, you can simply enter the new container name in the CONTAINER field and press Enter.

Figure 3-35. Confirm that Container Data Satisfied Enhanced Trap Criteria



```
---------------- XPEDITER/CICS - BROWSE CONTAINER DATA (2.C.3) ------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE: CWDEMPCH CSECT: CWDEMPCH          COMPILED: 30 JUN 2005 - 08:06:30

TASK:    00140                   PROGRAM:   CWDEMPCH
CHANNEL: PCHCHANNEL              CONTAINER: PCHREPCOMMAREA
DEC-OFFSET: 000000 ADD-OFFSET: 000000   CONTAINER-LENGTH: 0000000080
----+---10----+---20----+---30----+---40----+---50----+---60----+---70----+--->
Commarea: Text:TEST Char:Test Ascii:<(>? Hex: Test
----+---10----+---20----+---30----+---40----+---50----+---60----+---70----+--->
```

10. End your debugging session by typing **XPND** on a blank CICS screen and pressing Enter.

Setting an Enhanced Trap Using MQ

See the *Xpediter/CICS COBOL User's Guide* sections entitled “Using Enhanced Traps” and “Using Enhanced Breakpoints” for information on WebSphere MQ (MQSeries) traps and breakpoints.

Optimization Considerations

PL/I programs that are optimized using compiler options or using the CA-OPTIMIZER® compiler can be symbolically debugged with Xpediter/CICS. Depending on the optimizing algorithm employed, execution trace, code stepping, and resuming execution at another statement under Xpediter/CICS can appear to be incorrect.

The optimization technique used by high level language compilers attempts to improve the run-time performance of application programs. The methods typically used involve the rearrangement of object code to the point where there may be little correspondence between the sequence of generated machine instructions and the sequence of the source statements generated by the programmer.

Whether debugging is performed through specialized debugging software such as Xpediter/CICS, or through the conventional method of manually reading dumps and matching them to compile listings, the process is more complicated when optimization has been used. In circumstances where code has been relocated and/or re-sequenced by optimization, it can be much tougher to debug logic path problems.

Chapter 4. Debugging Applications Without Source Code

As discussed in "Preparing a Program for Execution" on page 2-1, the typical Xpediter/CICS test begins by processing the application program with the Compuware language processor. This process creates an online source listing that allows you to interactively step through your source code as it executes. However, if this source listing is not available, you can still test your programs with Xpediter/CICS using a form of testing called sourceless debugging.

This chapter demonstrates two methods for using sourceless debugging when testing your applications: PSEUDOSOURCE and Line Mode. You can still use Xpediter/CICS to set breakpoints, intercept abends, step through instructions, modify data, and generally test your program.

The PSEUDOSOURCE command allows you to map a program or CSECT to create an Assembler pseudo-listing. This pseudo-listing displays the underlying instructions in memory as if it was an Assembler program processed by the Compuware Assembler Language Processor. While you need some experience in debugging assembler language, the capability to debug a program or CSECT provides you with a much richer set of commands. It is important to understand that debugging a PL/I program using PSEUDOSOURCE is not a trivial undertaking. Having a compiler listing can make the task easier, but is not a requirement.

Line Mode debugging, although somewhat more complicated than using a pseudo-listing produced by the PSEUDOSOURCE command, still provides a set of commands that allows sourceless debugging of a PL/I program. Since you do not have a pseudo-listing available, you are restricted to using offsets and working with a single assembler instruction. In Line Mode, having a copy of the compiler listing is of tremendous help. You also need to decipher assembler instructions by looking at the hexadecimal display of storage that can be obtained from the 2.2 screen.

The following scenarios are based on the application transaction XPLE. This is the sample payroll transaction shipped with Xpediter/CICS that executes the PL/I program CWDEMPE. Since sourceless debugging mandates that no source listing exists for the program in question, you may wish to use a program of your own.

Note: The demonstrations in this chapter should be performed in Xpediter's standard operating mode — **not** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

When to Use Sourceless Debugging

Xpediter/CICS attempts to map your program to source whenever possible. It defaults to sourceless mode only if no corresponding source can be found. This is most likely due to one of the following situations:

- The program was not processed using a Compuware language processor.
- A version of the language processor no longer supported was used.
- The language processor step did not complete successfully.
- The source listing dataset is not defined (or incorrectly defined) to your CICS region.

- The load module timestamp does not match the Xpediter/CICS source timestamp because:
  - The program was linked into a library other than the one in use under CICS. Check your CICS library concatenation.
  - A new copy of the load module was not brought into storage.

Pressing the help PF key (default PF1) will display specific information on why no source is available.

Preparing for Sourceless Debugging

By definition, sourceless debugging only takes place if source does not exist in the Xpediter/CICS source listing dataset for a module. Program CWDEMPE has already been processed for use in other chapters of this guide. As a result, to continue with this section, you must first disable any Xpediter/CICS source listing for CWDEMPE. As an alternative, you could follow the examples using your own program that has not been processed with the language processor.

Type **=2.6.1** in the COMMAND field and **CWDEMPE** in the MODULE field, then press Enter to go to the List of CSECTs screen (2.6.1). Overtype CWDEMPE in the LISTING column with XXDEMPE and press Enter to disable the source listing. Be sure to change the name back to CWDEMPE when you have completed the procedures in this chapter, or simply end your Xpediter session.

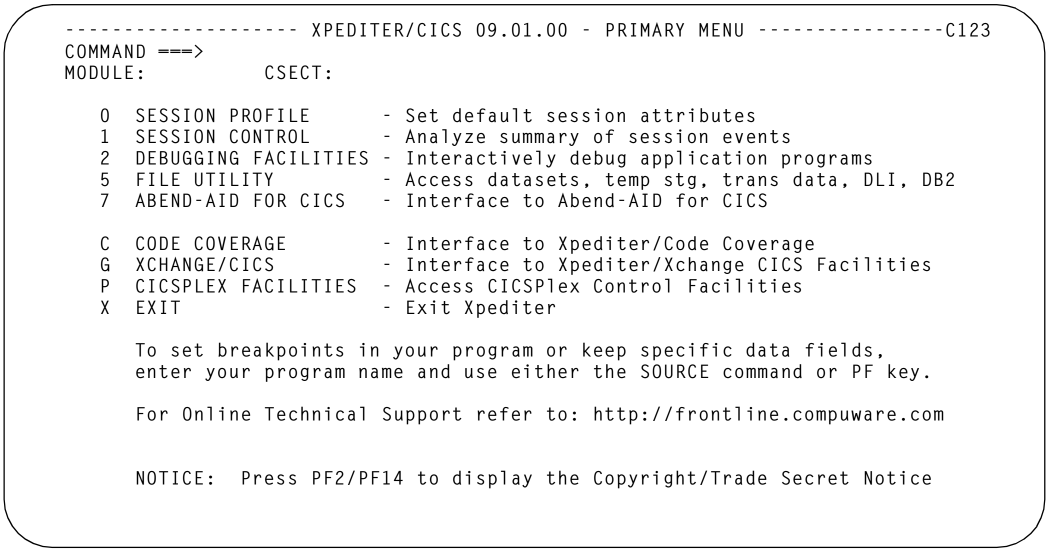
To follow the examples in this chapter you will also need to refer to a hardcopy listing of CWDEMPE that contains the Assembler listing and a copy of the Aggregate Length Table report.

Using PSEUDOSOURCE

Sourceless Debugging without Breakpoints

1. On a blank screen, type **XPED** and press Enter. This activates Xpediter/CICS for your terminal, turns on the abend trap option, and displays the Primary Menu as show in Figure 4-1.

Figure 4-1. Primary Menu (XPED/XPRT)



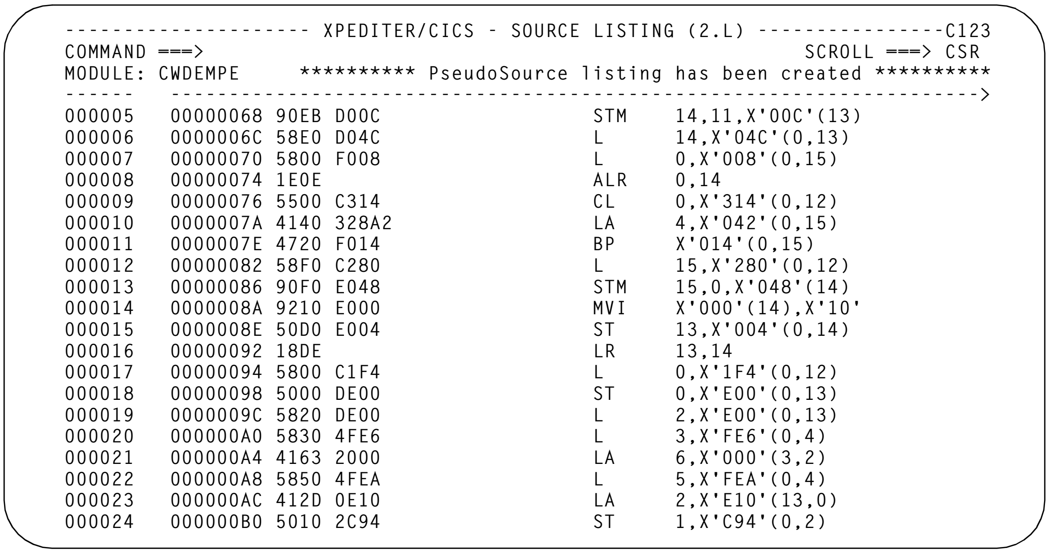
```
-------------------- XPEDITER/CICS 09.01.00 - PRIMARY MENU ----------------C123
COMMAND ===>
MODULE:          CSECT:

   0  SESSION PROFILE      - Set default session attributes
   1  SESSION CONTROL      - Analyze summary of session events
   2  DEBUGGING FACILITIES - Interactively debug application programs
   5  FILE UTILITY         - Access datasets, temp stg, trans data, DLI, DB2
   7  ABEND-AID FOR CICS   - Interface to Abend-AID for CICS

   C  CODE COVERAGE        - Interface to Xpediter/Code Coverage
   G  XCHANGE/CICS         - Interface to Xpediter/Xchange CICS Facilities
   P  CICSPLEX FACILITIES  - Access CICSPlex Control Facilities
   X  EXIT                 - Exit Xpediter

      To set breakpoints in your program or keep specific data fields,
      enter your program name and use either the SOURCE command or PF key.

      For Online Technical Support refer to: http://frontline.compuware.com


      NOTICE:  Press PF2/PF14 to display the Copyright/Trade Secret Notice
```

Xpediter/CICS is now turned ON and ready to intercept any abends associated with your terminal and programs.

2. To enable the display of the OFFSET field in Assembler programs, type **SET JUST OFF** in the COMMAND field and press Enter.

3. Type **PSEUDOSOURCE CWDEMPE** in the COMMAND line and press Enter. This automatically transfers you to the Source Listing screen (2.L) and displays the pseudo-listing created. This is shown in Figure 4-2. Press Clear to return to CICS to start your test.

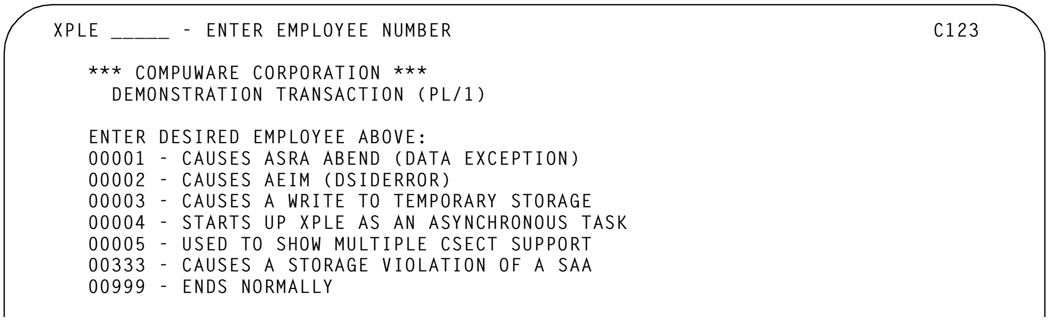
Figure 4-2. Source Listing Screen (2.L) with CWDEMPE Pseudo-Listing



```
--------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                     SCROLL ===> CSR
MODULE: CWDEMPE     ********** PseudoSource listing has been created **********
------   ----------------------------------------------------------------------->
000005   00000068 90EB D00C                     STM    14,11,X'00C'(13)
000006   0000006C 58E0 D04C                     L      14,X'04C'(0,13)
000007   00000070 5800 F008                     L      0,X'008'(0,15)
000008   00000074 1E0E                          ALR    0,14
000009   00000076 5500 C314                     CL     0,X'314'(0,12)
000010   0000007A 4140 328A2                    LA     4,X'042'(0,15)
000011   0000007E 4720 F014                     BP     X'014'(0,15)
000012   00000082 58F0 C280                     L      15,X'280'(0,12)
000013   00000086 90F0 E048                     STM    15,0,X'048'(14)
000014   0000008A 9210 E000                     MVI    X'000'(14),X'10'
000015   0000008E 50D0 E004                     ST     13,X'004'(0,14)
000016   00000092 18DE                          LR     13,14
000017   00000094 5800 C1F4                     L      0,X'1F4'(0,12)
000018   00000098 5000 DE00                     ST     0,X'E00'(0,13)
000019   0000009C 5820 DE00                     L      2,X'E00'(0,13)
000020   000000A0 5830 4FE6                     L      3,X'FE6'(0,4)
000021   000000A4 4163 2000                     LA     6,X'000'(3,2)
000022   000000A8 5850 4FEA                     L      5,X'FEA'(0,4)
000023   000000AC 412D 0E10                     LA     2,X'E10'(13,0)
000024   000000B0 5010 2C94                     ST     1,X'C94'(0,2)
```

4. On a blank CICS screen, type **XPLE** or your transaction ID and press Enter. This displays the Demonstration Transaction screen shown in Figure 4-3.

Figure 4-3. Demonstration Transaction Screen

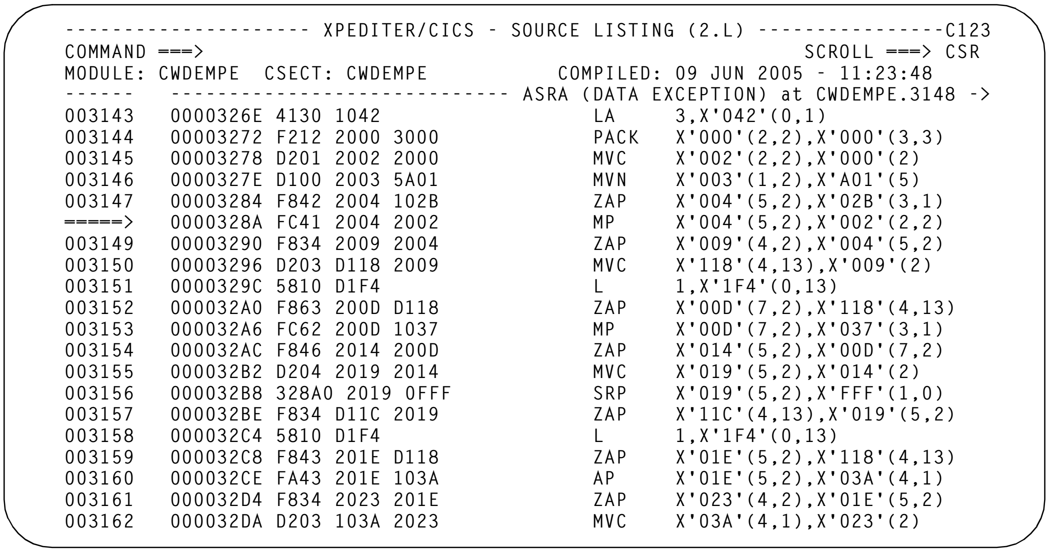


```
XPLE _____ - ENTER EMPLOYEE NUMBER                                        C123

   *** COMPUWARE CORPORATION ***
     DEMONSTRATION TRANSACTION (PL/1)

   ENTER DESIRED EMPLOYEE ABOVE:
   00001 - CAUSES ASRA ABEND (DATA EXCEPTION)
   00002 - CAUSES AEIM (DSIDERROR)
   00003 - CAUSES A WRITE TO TEMPORARY STORAGE
   00004 - STARTS UP XPLE AS AN ASYNCHRONOUS TASK
   00005 - USED TO SHOW MULTIPLE CSECT SUPPORT
   00333 - CAUSES A STORAGE VIOLATION OF A SAA
   00999 - ENDS NORMALLY
```

5. To cause an ASRA abend in CWDEMPE, type **00001** and press Enter.

   When you use the PSEUDOSOURCE command for CWDEMPE, it creates a pseudo-listing that appears to Xpediter/CICS as a program that was processed with the Assembler Language Processor. In this case, the Source Listing screen is displayed as shown in Figure 4-4. You can see the resulting abend was an ASRA, a data exception that occurred at statement 3148 in the pseudo-listing. The ASRA occurred on a multiply decimal (MP) instruction.

Figure 4-4. Source Listing Screen (2.L)



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ---------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48
------   ---------------------------- ASRA (DATA EXCEPTION) at CWDEMPE.3148 ->
003143   0000326E 4130 1042                  LA     3,X'042'(0,1)
003144   00003272 F212 2000 3000             PACK   X'000'(2,2),X'000'(3,3)
003145   00003278 D201 2002 2000             MVC    X'002'(2,2),X'000'(2)
003146   0000327E D100 2003 5A01             MVN    X'003'(1,2),X'A01'(5)
003147   00003284 F842 2004 102B             ZAP    X'004'(5,2),X'02B'(3,1)
=====>   0000328A FC41 2004 2002             MP     X'004'(5,2),X'002'(2,2)
003149   00003290 F834 2009 2004             ZAP    X'009'(4,2),X'004'(5,2)
003150   00003296 D203 D118 2009             MVC    X'118'(4,13),X'009'(2)
003151   0000329C 5810 D1F4                  L      1,X'1F4'(0,13)
003152   000032A0 F863 200D D118             ZAP    X'00D'(7,2),X'118'(4,13)
003153   000032A6 FC62 200D 1037             MP     X'00D'(7,2),X'037'(3,1)
003154   000032AC F846 2014 200D             ZAP    X'014'(5,2),X'00D'(7,2)
003155   000032B2 D204 2019 2014             MVC    X'019'(5,2),X'014'(2)
003156   000032B8 328A0 2019 0FFF            SRP    X'019'(5,2),X'FFF'(1,0)
003157   000032BE F834 D11C 2019             ZAP    X'11C'(4,13),X'019'(5,2)
003158   000032C4 5810 D1F4                  L      1,X'1F4'(0,13)
003159   000032C8 F843 201E D118             ZAP    X'01E'(5,2),X'118'(4,13)
003160   000032CE FA43 201E 103A             AP     X'01E'(5,2),X'03A'(4,1)
003161   000032D4 F834 2023 201E             ZAP    X'023'(4,2),X'01E'(5,2)
003162   000032DA D203 103A 2023             MVC    X'03A'(4,1),X'023'(2)
```

Since an ASRA took place on an MP instruction, there is a good chance that one or both of the multipliers contain invalid packed data. To test this theory, you can view the data. First, look at the instruction MP X'004'(5,2),X'002'(2,2). Since an MP instruction uses an IBM SS format, you can determine the first operand is 5 bytes long and is located X'004' off register 2. The second operand is 2 bytes long and located at X'002' off register 2. However, as we will see later, these offsets may not actually point to the fields in error. The next step is to find out where the data fields being multiplied originated.

Notice the interrupt offset is x'328A'. In order to determine the statement associated with the abend, we need to refer to the hardcopy of the compiler listing to locate the file line number containing (or nearest to) offset X'328A' (Figure 4-5). In our example, the LIST compiler option was used, so we need to refer to the generated assembler listing section of the listing. Look for the closest offset that is less than the interrupt offset **and** matches the MP instruction object code, which is FC41 2004 2002.

In our example, line number 837 contains a MP statement at offset x'324A'. It exactly matches the object code where the abend occurred. Therefore the line number we will look for in the source listing is 837.

Before continuing, it is important to note that while the generated pseudo-listing shows the ASRA at x'328A', the compiler listing shows the assembler instruction at x'324A'. This is a difference of x'40'. This difference is related to the method Xpediter/CICS uses to calculate the start of the executable code, and the way it is displayed in the compiled listing.

It is also important to note the first offset associated with line 837 is x'3222'. When resuming in high-level languages, Xpediter/CICS automatically resumes at the start of a statement, not the abend point. When using Sourceless debugging however, it is your responsibility to specify the correct resume offset. For right now, let's calculate the resume offset. This can be done very easily using the CALC command.

6. To determine our new resume offset type **?? +3222+40** in the COMMAND field and press Enter. The resulting value is x'3262'. Remember this value for later. You are probably asking, "why add 40?" It is the difference determined in the previous paragraph and the adjustment factor we need to make.

Figure 4-5. Assembler Listing (LIST) from COBOL Compiler

```
003222                     00836 |        @1L79    DS     0H
003222  4820  4FCA         00837 |                 LH     r2,=H'1'
003226  4122  D000         00837 |                 LA     r2,0(r2,r13)
00322A  5810  D1F4         00837 |                 L      r1,WA_PTR(,r13,500)
00322E  4130  1042         00837 |                 LA     r3,_shadow8(,r1,66)
003232  F212  2000  3000   00837 |                 PACK   #pdr3@3336_1(2,r2,0),_shadow8(3,r3,0)
003238  D201  2002  2000   00837 |                 MVC    #pdr3@3338_1(2,r2,2),#pdr3@3336_1(r2,0)
00323E  D100  2003  5A01   00837 |                 MVN    #pdr3@3338_1(1,r2,3),+CONSTANT_AREA(r5,2561)
003244  F842  2004  102B   00837 |                 ZAP    #pdr9@3340_1(5,r2,4),_shadow4(3,r1,43)
00324A  FC41  2004  2002   00837 |                 MP     #pdr9@3340_1(5,r2,4),#pdr3@3338_1(2,r2,2)
003250  F834  2009  2004   00837 |                 ZAP    #pdr7@3345_1(4,r2,9),#pdr9@3340_1(5,r2,4)
003256  D203  D118  2009   00837 |                 MVC    CURR_PAY(4,r13,280),#pdr7@3345_1(r2,9)
00325C  5810  D1F4         00838 |                 L      r1,WA_PTR(,r13,500)
```

If you find line number 837 in the Procedure Division of your listing (Figure 4-6), you'll notice the program is trying to multiply WA_HOURS by WA_RATE. One or both of these fields contains invalid data that caused the exception.

Figure 4-6. Source Listing from Compiler Listing

```
  832.1      L300_EMPLOYEE_PAY_RTN:                                                00000761
  833.1                                                                            00000762
  834.1         SELECT(WA_TYPE);                                                   00000763
  835.1             WHEN('N','I','S')                                              00000764
  836.1                  DO;                                                       00000765
  837.1                       CURR_PAY   =  WA_HOURS * WA_RATE ;                   00000766
  838.1                       CURR_TAXES =  CURR_PAY * WA_TAX_RAT ;                00000767
  839.1                       WA_YTD_GRS = CURR_PAY + WA_YTD_GRS ;                 00000768
  840.1                       WA_YTD_TAX = CURR_TAXES +  WA_YTD_TAX ;              00000769
  841.1                  END ;                                                     00000770
  842.1              OTHERWISE;                                                    00000771
  843.1         END;                                                               00000772
```

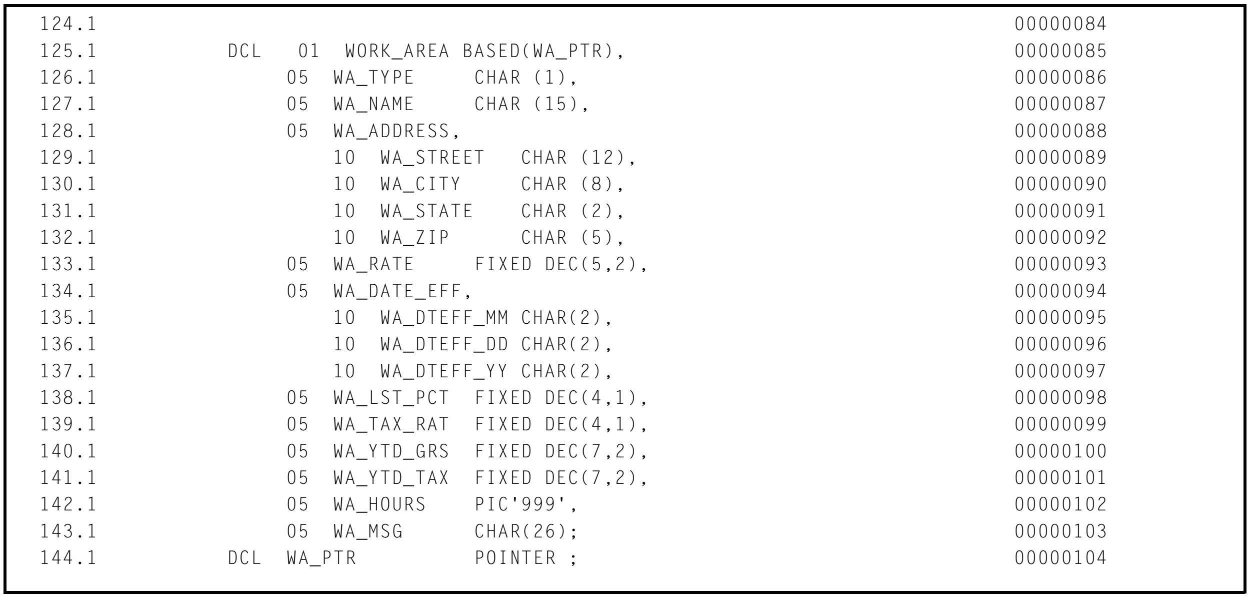
Modifying Storage

Next, you will apply the values from the listing to view and modify the data in storage.

Before continuing, we know the code generated by the compiler isn't straightforward and simple. In looking at the code for line number 837 (Figure 4-5 on page 4-5), we see there are no references to the variables WA_HOURS or WA_RATE. This probably means the actual MP instruction where the abend occurred does not necessarily directly reference the variables either. In order to verify where the variables reside, we need to reference our compiled listing.

If we look in the source listing where WA_HOURS and WA_RATE are defined, we discover they are actually defined in a structure 'WORK_AREA', which is a based variable (Figure 4-7). It is based on field WA_PTR. The value in field WA_PTR is the actual address of the structure.

Figure 4-7. Data Division from Compiler Listing



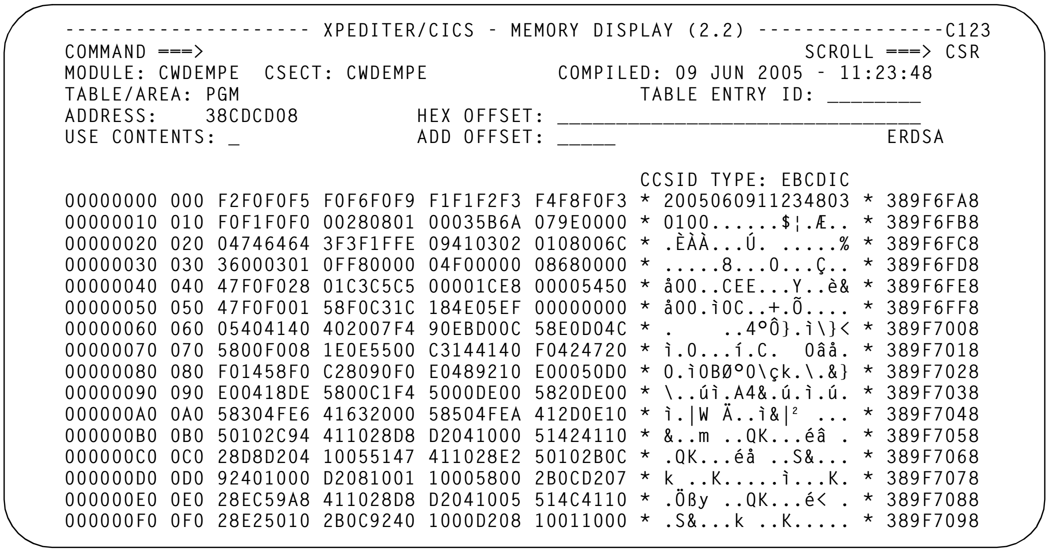
```
124.1                                                                        00000084
125.1          DCL   01  WORK_AREA BASED(WA_PTR),                            00000085
126.1               05  WA_TYPE     CHAR (1),                                00000086
127.1               05  WA_NAME     CHAR (15),                               00000087
128.1               05  WA_ADDRESS,                                          00000088
129.1                   10  WA_STREET   CHAR (12),                           00000089
130.1                   10  WA_CITY     CHAR (8),                            00000090
131.1                   10  WA_STATE    CHAR (2),                            00000091
132.1                   10  WA_ZIP      CHAR (5),                            00000092
133.1               05  WA_RATE     FIXED DEC(5,2),                          00000093
134.1               05  WA_DATE_EFF,                                         00000094
135.1                   10  WA_DTEFF_MM CHAR(2),                             00000095
136.1                   10  WA_DTEFF_DD CHAR(2),                             00000096
137.1                   10  WA_DTEFF_YY CHAR(2),                             00000097
138.1               05  WA_LST_PCT  FIXED DEC(4,1),                          00000098
139.1               05  WA_TAX_RAT  FIXED DEC(4,1),                          00000099
140.1               05  WA_YTD_GRS  FIXED DEC(7,2),                          00000100
141.1               05  WA_YTD_TAX  FIXED DEC(7,2),                          00000101
142.1               05  WA_HOURS    PIC'999',                                00000102
143.1               05  WA_MSG      CHAR(26);                                00000103
144.1          DCL  WA_PTR          POINTER ;                                00000104
```

We need to determine the address currently in WA_PTR. First, we need to find the offset of WA_PTR in the DSA for the program. Unfortunately, we do not have a variable storage map section in our listing. While this would make it easy to find where WA-PTR resides, we can simply look in the generated code. If we look at Figure 4-5 on page 4-5, we can see that WA_PTR is referenced in the instruction 'L r1,WA_PTR(,r13,500)'. The comment indicates it is at offset 500 (x'1F4') from register 13.

To find the field containing the invalid data, we first locate to the storage pointed at by register 13. The Memory Display screen (2.2) is used to locate the invalid data.

1. Press PF14, or type =**2.2** in the COMMAND field and press Enter. The Memory Display screen (2.2) as shown in Figure 4-8 appears.

Figure 4-8. Viewing Program Storage on the Memory Display Screen (2.2)

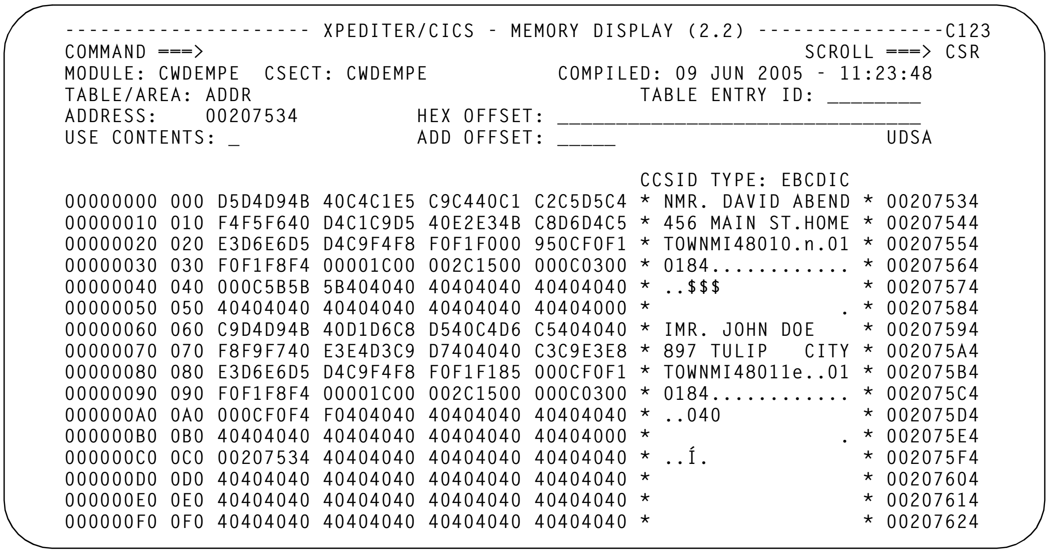


```
-------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ----------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
TABLE/AREA: PGM                                  TABLE ENTRY ID: ________
ADDRESS:    38CDCD08          HEX OFFSET: ________________________________
USE CONTENTS: _               ADD OFFSET: _____                     ERDSA

                                                 CCSID TYPE: EBCDIC
00000000 000 F2F0F0F5 F0F6F0F9 F1F1F2F3 F4F8F0F3 * 2005060911234803 * 389F6FA8
00000010 010 F0F1F0F0 00280801 00035B6A 079E0000 * 0100......$¦.Æ.. * 389F6FB8
00000020 020 04746464 3F3F1FFE 09410302 0108006C * .ÈÀÀ...Ú. .....% * 389F6FC8
00000030 030 36000301 0FF80000 04F00000 08680000 * .....8...0...Ç.. * 389F6FD8
00000040 040 47F0F028 01C3C5C5 00001CE8 00005450 * å00..CEE...Y..è& * 389F6FE8
00000050 050 47F0F001 58F0C31C 184E05EF 00000000 * å00.ì0C..+.Õ.... * 389F6FF8
00000060 060 05404140 402007F4 90EBD00C 58E0D04C * .    ..4°Ô}.ì\}< * 389F7008
00000070 070 5800F008 1E0E5500 C3144140 F0424720 * ì.0...í.C.  0âå. * 389F7018
00000080 080 F01458F0 C28090F0 E0489210 E00050D0 * 0.ì0BØ°0\çk.\.&} * 389F7028
00000090 090 E00418DE 5800C1F4 5000DE00 5820DE00 * \..úì.A4&.ú.ì.ú. * 389F7038
000000A0 0A0 58304FE6 41632000 58504FEA 412D0E10 * ì.|W Ä..ì&|²  ... * 389F7048
000000B0 0B0 50102C94 411028D8 D2041000 51424110 * &..m ..QK...éâ . * 389F7058
000000C0 0C0 28D8D204 10055147 411028E2 50102B0C * .QK...éå ..S&... * 389F7068
000000D0 0D0 92401000 D2081001 10005800 2B0CD207 * k ..K.....ì...K. * 389F7078
000000E0 0E0 28EC59A8 411028D8 D2041005 514C4110 * .Ößy ..QK...é< . * 389F7088
000000F0 0F0 28E25010 2B0C9240 1000D208 10011000 * .S&...k ..K..... * 389F7098
```

The Memory Display screen (2.2) shows a hexadecimal dump of the your test program. Any instruction or data in your program is available from here. Simply type the displacement you wish to see in the HEX (or ADD) OFFSET fields.

2. To display storage for the work area containing the bad data (WA_PTR), overtype PGM in the TABLE/AREA field with **R13**, type **+1F4** in the HEX OFFSET field, type **X** in the USE CONTENTS field and press Enter. This positions the display to the address contained in field WA_PTR. Figure 4-9 shows the storage for WORK_AREA displayed on the Memory Display screen (2.2).

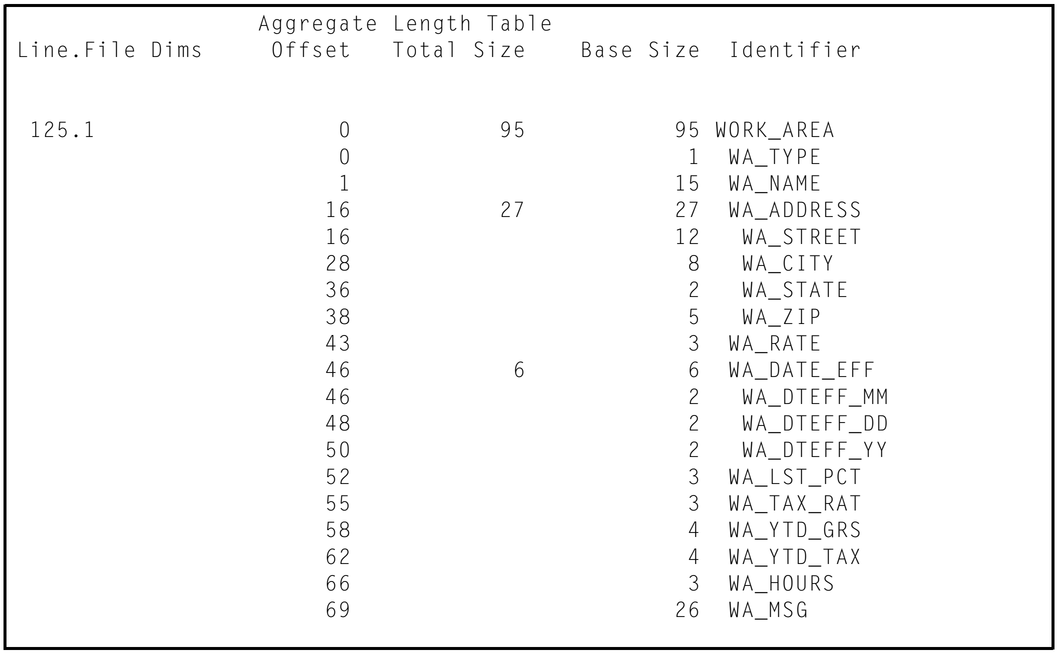
Figure 4-9. Displaying the Address of WA_PTR



```
-------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ----------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
TABLE/AREA: ADDR                                 TABLE ENTRY ID: ________
ADDRESS:    00207534          HEX OFFSET: ________________________________
USE CONTENTS: _               ADD OFFSET: _____                     UDSA

                                                 CCSID TYPE: EBCDIC
00000000 000 D5D4D94B 40C4C1E5 C9C440C1 C2C5D5C4 * NMR. DAVID ABEND * 00207534
00000010 010 F4F5F640 D4C1C9D5 40E2E34B C8D6D4C5 * 456 MAIN ST.HOME * 00207544
00000020 020 E3D6E6D5 D4C9F4F8 F0F1F000 950CF0F1 * TOWNMI48010.n.01 * 00207554
00000030 030 F0F1F8F4 00001C00 002C1500 000C0300 * 0184............ * 00207564
00000040 040 000C5B5B 5B404040 40404040 40404040 * ..$$$            * 00207574
00000050 050 40404040 40404040 40404040 40404000 *                . * 00207584
00000060 060 C9D4D94B 40D1D6C8 D540C4D6 C5404040 * IMR. JOHN DOE    * 00207594
00000070 070 F8F9F740 E3E4D3C9 D7404040 C3C9E3E8 * 897 TULIP   CITY * 002075A4
00000080 080 E3D6E6D5 D4C9F4F8 F0F1F185 000CF0F1 * TOWNMI48011e..01 * 002075B4
00000090 090 F0F1F8F4 00001C00 002C1500 000C0300 * 0184............ * 002075C4
000000A0 0A0 000CF0F4 F0404040 40404040 40404040 * ..040            * 002075D4
000000B0 0B0 40404040 40404040 40404040 40404000 *                . * 002075E4
000000C0 0C0 00207534 40404040 40404040 40404040 * ..Í.             * 002075F4
000000D0 0D0 40404040 40404040 40404040 40404040 *                  * 00207604
000000E0 0E0 40404040 40404040 40404040 40404040 *                  * 00207614
000000F0 0F0 40404040 40404040 40404040 40404040 *                  * 00207624
```

From the Aggregate Length table (condensed from the actual listing) shown in Figure 4-10, find the WORK_AREA structure. The structure shows the offsets and lengths of WA_RATE (3 bytes, offset 43) and WA_HOURS (3 bytes, offset 66).

Figure 4-10. Aggregate Length Table



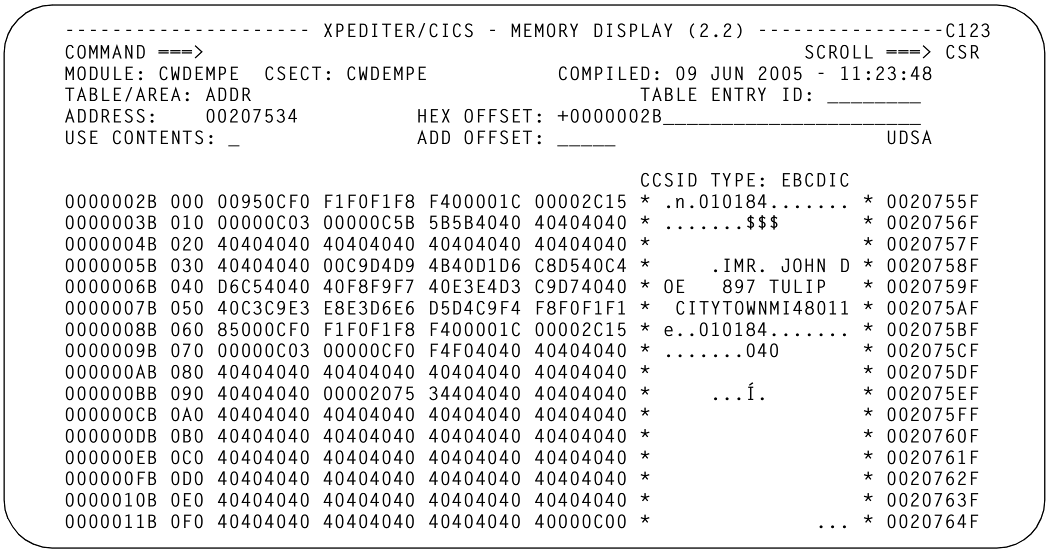
```
                  Aggregate Length Table
Line.File Dims     Offset   Total Size    Base Size  Identifier


 125.1                  0           95           95 WORK_AREA
                        0                         1  WA_TYPE
                        1                        15  WA_NAME
                       16           27           27  WA_ADDRESS
                       16                        12   WA_STREET
                       28                         8   WA_CITY
                       36                         2   WA_STATE
                       38                         5   WA_ZIP
                       43                         3  WA_RATE
                       46            6            6  WA_DATE_EFF
                       46                         2   WA_DTEFF_MM
                       48                         2   WA_DTEFF_DD
                       50                         2   WA_DTEFF_YY
                       52                         3  WA_LST_PCT
                       55                         3  WA_TAX_RAT
                       58                         4  WA_YTD_GRS
                       62                         4  WA_YTD_TAX
                       66                         3  WA_HOURS
                       69                        26  WA_MSG
```

Notice the offsets are shown in decimal format. To convert them to hex format, use the CALC command as follows:

- On the COMMAND line, type **?? 43** and press Enter to convert the offset for WA_RATE. The result is 2B in hex format.
- To find the hex offset of WA_HOURS, type **?? 66** and press Enter. The result is 42.

3. To display the contents of your first multiplier (WA_RATE), tell Xpediter/CICS where to look by typing **+2B** in the HEX OFFSET field. When you press Enter, the display is positioned at the first data item as shown in Figure 4-11.

Figure 4-11. Viewing Data Fields on the Memory Display Screen (2.2)

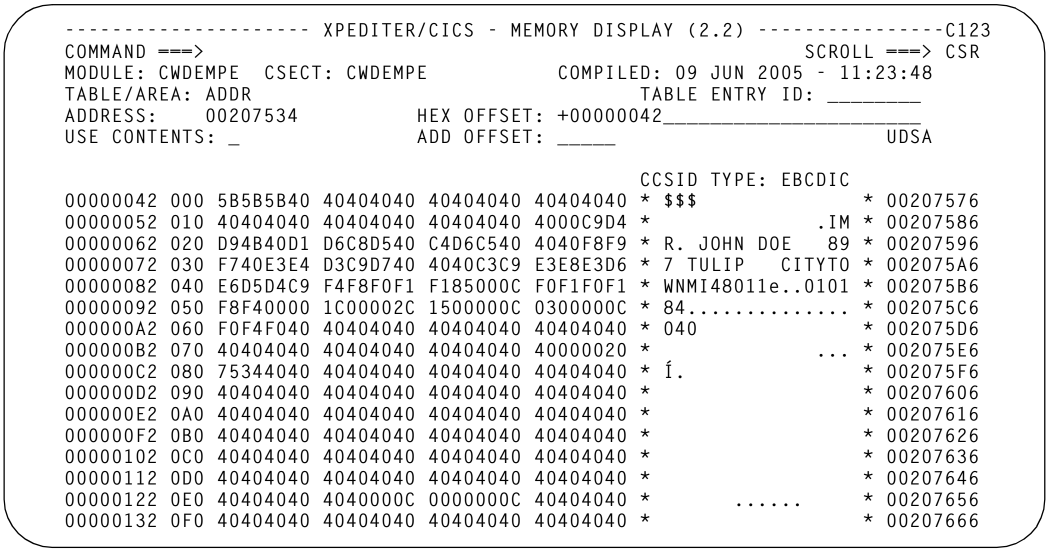


```
-------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ---------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48
TABLE/AREA: ADDR                                TABLE ENTRY ID: ________
ADDRESS:    00207534          HEX OFFSET: +0000002B______________________
USE CONTENTS: _               ADD OFFSET: _____                      UDSA

                                                CCSID TYPE: EBCDIC
0000002B 000 00950CF0 F1F0F1F8 F400001C 00002C15 * .n.010184....... * 0020755F
0000003B 010 00000C03 00000C5B 5B5B4040 40404040 * .......$$$       * 0020756F
0000004B 020 40404040 40404040 40404040 40404040 *                  * 0020757F
0000005B 030 40404040 00C9D4D9 4B40D1D6 C8D540C4 *     .IMR. JOHN D * 0020758F
0000006B 040 D6C54040 40F8F9F7 40E3E4D3 C9D74040 * OE   897 TULIP   * 0020759F
0000007B 050 40C3C9E3 E8E3D6E6 D5D4C9F4 F8F0F1F1 *  CITYTOWNMI48011 * 002075AF
0000008B 060 85000CF0 F1F0F1F8 F400001C 00002C15 * e..010184....... * 002075BF
0000009B 070 00000C03 00000CF0 F4F04040 40404040 * .......040       * 002075CF
000000AB 080 40404040 40404040 40404040 40404040 *                  * 002075DF
000000BB 090 40404040 00002075 34404040 40404040 *     ...Í.        * 002075EF
000000CB 0A0 40404040 40404040 40404040 40404040 *                  * 002075FF
000000DB 0B0 40404040 40404040 40404040 40404040 *                  * 0020760F
000000EB 0C0 40404040 40404040 40404040 40404040 *                  * 0020761F
000000FB 0D0 40404040 40404040 40404040 40404040 *                  * 0020762F
0000010B 0E0 40404040 40404040 40404040 40404040 *                  * 0020763F
0000011B 0F0 40404040 40404040 40404040 40000C00 *              ... * 0020764F
```

Note the displayed data area contains valid packed information (x'00950C').

4. Display the contents of the second data item (WA_HOURS) by typing **+42** in the cleared HEX OFFSET field. When you press Enter, the screen is positioned at the second data item as shown in Figure 4-12.

Figure 4-12. WA_HOURS Data Field on the Memory Display Screen (2.2)

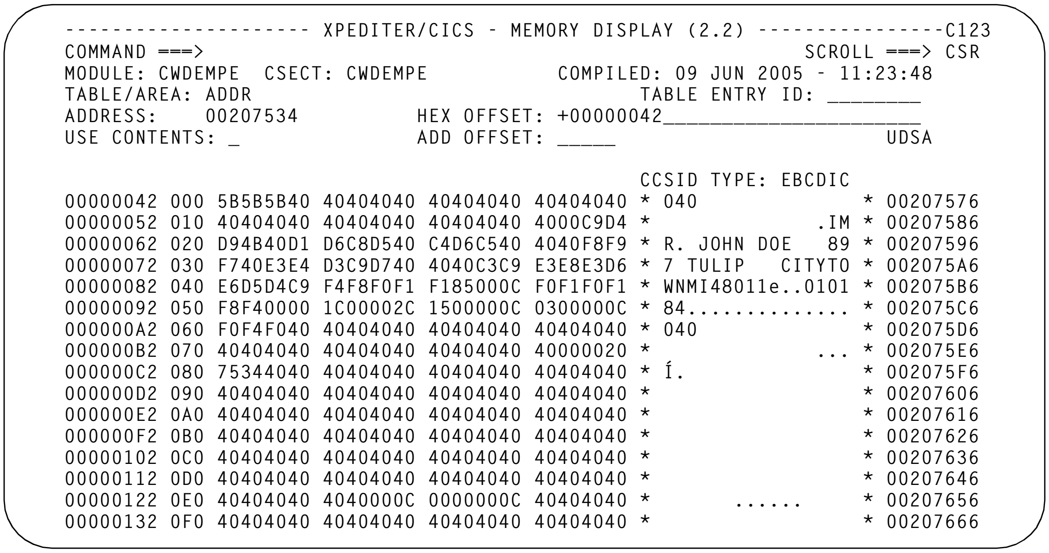


```
--------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ---------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
TABLE/AREA: ADDR                                 TABLE ENTRY ID: ________
ADDRESS:    00207534          HEX OFFSET: +00000042_______________________
USE CONTENTS: _               ADD OFFSET: _____                     UDSA

                                                 CCSID TYPE: EBCDIC
00000042 000 5B5B5B40 40404040 40404040 40404040 * $$$              * 00207576
00000052 010 40404040 40404040 40404040 4000C9D4 *              .IM * 00207586
00000062 020 D94B40D1 D6C8D540 C4D6C540 4040F8F9 * R. JOHN DOE   89 * 00207596
00000072 030 F740E3E4 D3C9D740 4040C3C9 E3E8E3D6 * 7 TULIP   CITYTO * 002075A6
00000082 040 E6D5D4C9 F4F8F0F1 F185000C F0F1F0F1 * WNMI48011e..0101 * 002075B6
00000092 050 F8F40000 1C00002C 1500000C 0300000C * 84.............. * 002075C6
000000A2 060 F0F4F040 40404040 40404040 40404040 * 040              * 002075D6
000000B2 070 40404040 40404040 40404040 40000020 *              ... * 002075E6
000000C2 080 75344040 40404040 40404040 40404040 * Í.               * 002075F6
000000D2 090 40404040 40404040 40404040 40404040 *                  * 00207606
000000E2 0A0 40404040 40404040 40404040 40404040 *                  * 00207616
000000F2 0B0 40404040 40404040 40404040 40404040 *                  * 00207626
00000102 0C0 40404040 40404040 40404040 40404040 *                  * 00207636
00000112 0D0 40404040 40404040 40404040 40404040 *                  * 00207646
00000122 0E0 40404040 4040000C 0000000C 40404040 *       ......     * 00207656
00000132 0F0 40404040 40404040 40404040 40404040 *                  * 00207666
```

This time the displayed field contains bad data. The three-byte field contains 5B5B5B, otherwise known as $$$. This error caused the ASRA.

5. Position the cursor on the bad data (**$$$**).
6. Replace the bad data by typing a valid number, such as **040**, over the data.
7. Press Enter to update the change. The area contains valid data as shown in Figure 4-13.

Figure 4-13. Modifying Data on the Memory Display Screen (2.2)

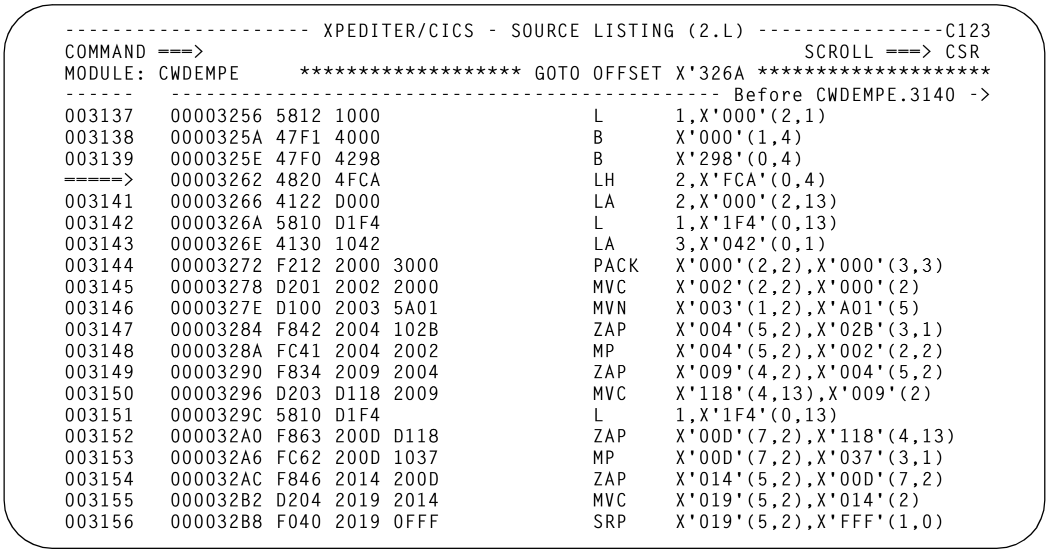


```
--------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ---------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
TABLE/AREA: ADDR                                 TABLE ENTRY ID: ________
ADDRESS:    00207534          HEX OFFSET: +00000042_______________________
USE CONTENTS: _               ADD OFFSET: _____                     UDSA

                                                 CCSID TYPE: EBCDIC
00000042 000 5B5B5B40 40404040 40404040 40404040 * 040              * 00207576
00000052 010 40404040 40404040 40404040 4000C9D4 *              .IM * 00207586
00000062 020 D94B40D1 D6C8D540 C4D6C540 4040F8F9 * R. JOHN DOE   89 * 00207596
00000072 030 F740E3E4 D3C9D740 4040C3C9 E3E8E3D6 * 7 TULIP   CITYTO * 002075A6
00000082 040 E6D5D4C9 F4F8F0F1 F185000C F0F1F0F1 * WNMI48011e..0101 * 002075B6
00000092 050 F8F40000 1C00002C 1500000C 0300000C * 84.............. * 002075C6
000000A2 060 F0F4F040 40404040 40404040 40404040 * 040              * 002075D6
000000B2 070 40404040 40404040 40404040 40000020 *              ... * 002075E6
000000C2 080 75344040 40404040 40404040 40404040 * Í.               * 002075F6
000000D2 090 40404040 40404040 40404040 40404040 *                  * 00207606
000000E2 0A0 40404040 40404040 40404040 40404040 *                  * 00207616
000000F2 0B0 40404040 40404040 40404040 40404040 *                  * 00207626
00000102 0C0 40404040 40404040 40404040 40404040 *                  * 00207636
00000112 0D0 40404040 40404040 40404040 40404040 *                  * 00207646
00000122 0E0 40404040 4040000C 0000000C 40404040 *       ......     * 00207656
00000132 0F0 40404040 40404040 40404040 40404040 *                  * 00207666
```

Now that you have corrected the data causing the error, you can continue with the test.

8. Type **=2.L** in the COMMAND field and press Enter to return to the Source Listing screen.
9. To position to the beginning of the actual statement, type **GOTO +3262** (or **GOTO 3140**) in the COMMAND line and press Enter. This repositions the execution pointer to the beginning of the actual PL/I statement as shown in Figure 4-14.

Figure 4-14. Source Listing Screen (2.L) after using GOTO +3262 Command

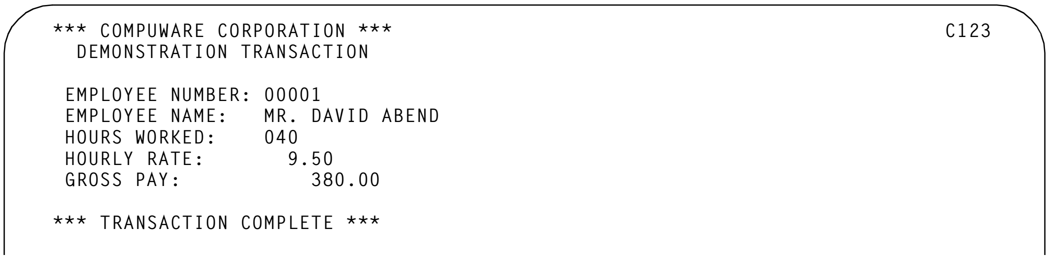


```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE: CWDEMPE     ******************* GOTO OFFSET X'326A ********************
------   --------------------------------------------- Before CWDEMPE.3140 ->
003137   00003256 5812 1000                L      1,X'000'(2,1)
003138   0000325A 47F1 4000                B      X'000'(1,4)
003139   0000325E 47F0 4298                B      X'298'(0,4)
=====>   00003262 4820 4FCA                LH     2,X'FCA'(0,4)
003141   00003266 4122 D000                LA     2,X'000'(2,13)
003142   0000326A 5810 D1F4                L      1,X'1F4'(0,13)
003143   0000326E 4130 1042                LA     3,X'042'(0,1)
003144   00003272 F212 2000 3000           PACK   X'000'(2,2),X'000'(3,3)
003145   00003278 D201 2002 2000           MVC    X'002'(2,2),X'000'(2)
003146   0000327E D100 2003 5A01           MVN    X'003'(1,2),X'A01'(5)
003147   00003284 F842 2004 102B           ZAP    X'004'(5,2),X'02B'(3,1)
003148   0000328A FC41 2004 2002           MP     X'004'(5,2),X'002'(2,2)
003149   00003290 F834 2009 2004           ZAP    X'009'(4,2),X'004'(5,2)
003150   00003296 D203 D118 2009           MVC    X'118'(4,13),X'009'(2)
003151   0000329C 5810 D1F4                L      1,X'1F4'(0,13)
003152   000032A0 F863 200D D118           ZAP    X'00D'(7,2),X'118'(4,13)
003153   000032A6 FC62 200D 1037           MP     X'00D'(7,2),X'037'(3,1)
003154   000032AC F846 2014 200D           ZAP    X'014'(5,2),X'00D'(7,2)
003155   000032B2 D204 2019 2014           MVC    X'019'(5,2),X'014'(2)
003156   000032B8 F040 2019 0FFF           SRP    X'019'(5,2),X'FFF'(1,0)
```

10. To complete your test, re-execute line 837 by press PF12, or typing **GO** in the COMMAND line and pressing Enter. This time the transaction should not abend.

    The program finishes without further abends, concluding our exercise in sourceless debugging (Figure 4-15).

Figure 4-15. Demonstration Transaction Screen



```
*** COMPUWARE CORPORATION ***                                              C123
  DEMONSTRATION TRANSACTION

 EMPLOYEE NUMBER: 00001
 EMPLOYEE NAME:   MR. DAVID ABEND
 HOURS WORKED:    040
 HOURLY RATE:       9.50
 GROSS PAY:           380.00

*** TRANSACTION COMPLETE ***
```

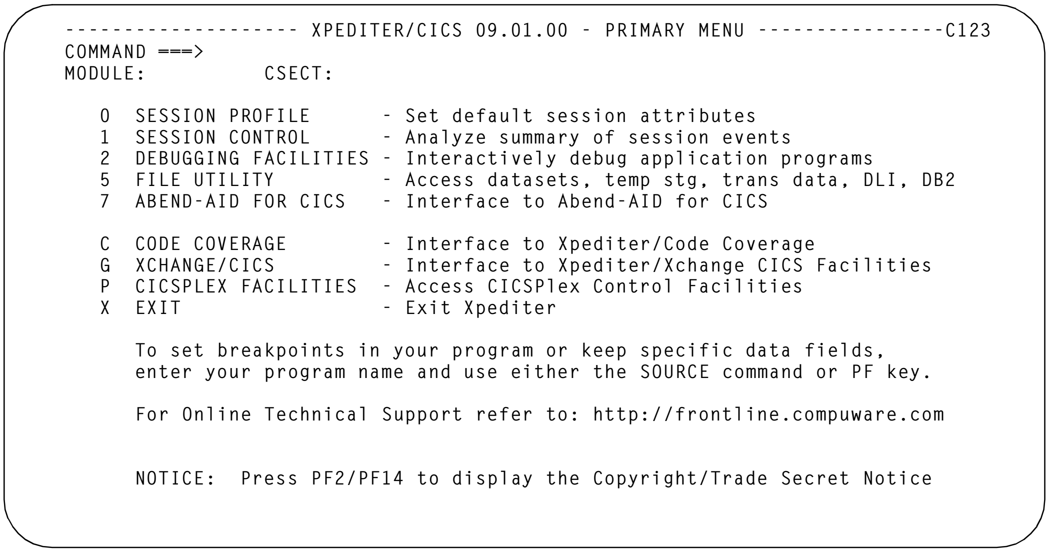
11. Press Clear to return to CICS.
12. On a blank CICS screen, type **XPND** and press Enter to end your debugging session.

Line Mode

Sourceless Debugging without Breakpoints

1. On a blank screen, type **XPED** and press Enter. This activates Xpediter/CICS for your terminal, turns on the abend trap option, and displays the Primary Menu as shown in Figure 4-16.

Figure 4-16. Primary Menu (XPED/XPRT)

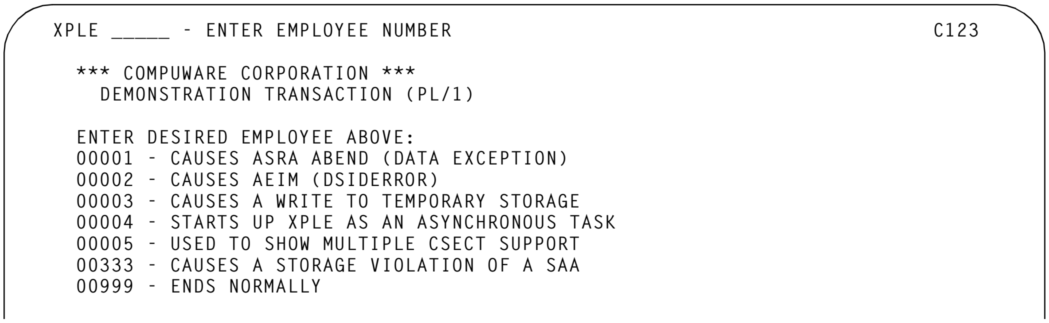


```
------------------- XPEDITER/CICS 09.01.00 - PRIMARY MENU ----------------C123
COMMAND ===>
MODULE:          CSECT:

   0  SESSION PROFILE      - Set default session attributes
   1  SESSION CONTROL      - Analyze summary of session events
   2  DEBUGGING FACILITIES - Interactively debug application programs
   5  FILE UTILITY         - Access datasets, temp stg, trans data, DLI, DB2
   7  ABEND-AID FOR CICS   - Interface to Abend-AID for CICS

   C  CODE COVERAGE        - Interface to Xpediter/Code Coverage
   G  XCHANGE/CICS         - Interface to Xpediter/Xchange CICS Facilities
   P  CICSPLEX FACILITIES  - Access CICSPlex Control Facilities
   X  EXIT                 - Exit Xpediter

      To set breakpoints in your program or keep specific data fields,
      enter your program name and use either the SOURCE command or PF key.

      For Online Technical Support refer to: http://frontline.compuware.com


      NOTICE:  Press PF2/PF14 to display the Copyright/Trade Secret Notice
```

Xpediter/CICS is now turned ON and ready to intercept any abends associated with your terminal and programs.

2. To turn the trace option on, type **SET TRACE ON** in the COMMAND field and press Enter.
3. Press Clear to return to CICS to start your test.
4. On a blank CICS screen, type **XPLE** or your transaction ID, and press Enter. This displays the Demonstration Transaction screen shown in Figure 4-17.

Figure 4-17. Demonstration Transaction Screen

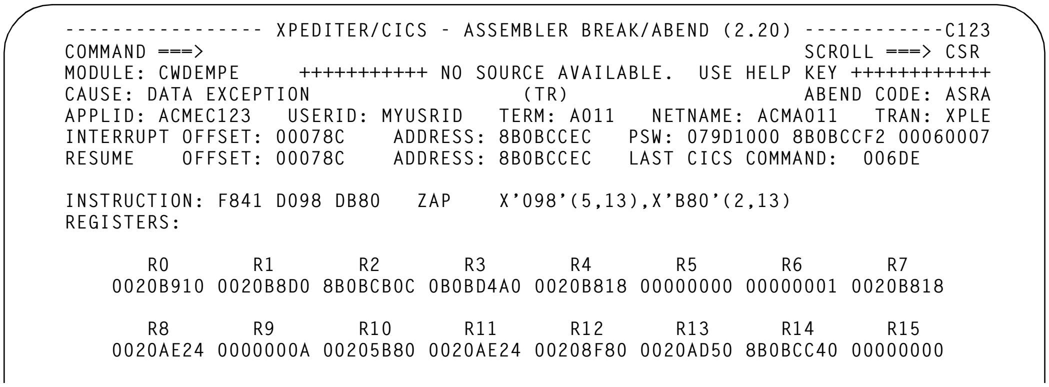


```
XPLE _____ - ENTER EMPLOYEE NUMBER                                      C123

  *** COMPUWARE CORPORATION ***
    DEMONSTRATION TRANSACTION (PL/1)

  ENTER DESIRED EMPLOYEE ABOVE:
  00001 - CAUSES ASRA ABEND (DATA EXCEPTION)
  00002 - CAUSES AEIM (DSIDERROR)
  00003 - CAUSES A WRITE TO TEMPORARY STORAGE
  00004 - STARTS UP XPLE AS AN ASYNCHRONOUS TASK
  00005 - USED TO SHOW MULTIPLE CSECT SUPPORT
  00333 - CAUSES A STORAGE VIOLATION OF A SAA
  00999 - ENDS NORMALLY
```

5. To cause an ASRA abend in CWDEMPE, type **00001** and press Enter.

   When no source is available for the abending program, Xpediter/CICS displays the Assembler Break/Abend screen (2.20) as shown in Figure 4-18.

Figure 4-18. Assembler Break/Abend Screen (2.20)



```
----------------- XPEDITER/CICS - ASSEMBLER BREAK/ABEND (2.20) ------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE     +++++++++++ NO SOURCE AVAILABLE.  USE HELP KEY ++++++++++++
CAUSE: DATA EXCEPTION                  (TR)                    ABEND CODE: ASRA
APPLID: ACMEC123   USERID: MYUSRID   TERM: A011   NETNAME: ACMA011   TRAN: XPLE
INTERRUPT OFFSET: 00078C    ADDRESS: 8B0BCCEC   PSW: 079D1000 8B0BCCF2 00060007
RESUME    OFFSET: 00078C    ADDRESS: 8B0BCCEC   LAST CICS COMMAND:  006DE

INSTRUCTION: F841 D098 DB80   ZAP    X'098'(5,13),X'B80'(2,13)
REGISTERS:

       R0       R1       R2       R3       R4       R5       R6       R7
    0020B910 0020B8D0 8B0BCB0C 0B0BD4A0 0020B818 00000000 00000001 0020B818

       R8       R9       R10      R11      R12      R13      R14      R15
    0020AE24 0000000A 00205B80 0020AE24 00208F80 0020AD50 8B0BCC40 00000000
```

The program name is displayed at the top of the screen. Below the program name is the cause of the abend, which is a data exception, and the abend code, which is ASRA. Statistical information such as the current offset of the interrupted instruction, its physical address, and the program status word (PSW) in use are displayed. Notice that Xpediter/CICS displays the resume offset and address, should you wish to continue processing.

Next we see the current instruction, complete with its operation code and operands. Xpediter/CICS also shows that disassembled instruction to the right. From this, you see the ASRA occurred on a Zero Add Pack (ZAP) instruction.

Toward the bottom of the screen, all 16 of the current general purpose registers are displayed.

Since an ASRA took place on a ZAP instruction, there is a good chance the second operand contains invalid packed data. To test this theory, you can view the data. But first, look at the instruction ZAP X'098'(5,13),X'B80'(2,13). Since a ZAP instruction uses an IBM SS format, you can determine the first operand is 5 bytes long and is located x'098' off register 13. The second operand is 2 bytes in length, located x'B80' off register 13.

In order to find the PL/I source statement in error, refer to your Assembler listing and locate the following:

- Interrupt offset
- Statement number
- Offset for the beginning of the statement (Once you fix the bad data, this offset will be your resume location.)

  In our example, the statement in error is statement number 91, which begins at offset x'77A'.

Figure 4-19. Assembler Listing

```
      90   1  1         WHEN('N','I','S')
                             DO;
      91   1  2                   CURR_PAY    =  WA_HOURS * WA_RATE ;
      92   1  2                   CURR_TAXES =  CURR_PAY * WA_TAX_RAT ;
      93   1  2                   WA_YTD_GRS = CURR_PAY + WA_YTD_GRS ;
      94   1  2                   WA_YTD_TAX = CURR_TAXES +  WA_YTD_TAX ;
.
.
.

* STATEMENT NUMBER  91
00077A  58 70 D 0CC               L     7,WA_PTR
00077E  F2 12 D B80 7 042         PACK  2944(2,13),WORK_AR
                                        EA.WA_HOURS(3)
000784  96 0F D B81               OI    2945(13),X'0F'
000788  94 FC D B81               NI    2945(13),X'FC'
00078C  F8 41 D 098 D B80         ZAP   WKSP.1+32(5),2944(
                                        2,13)
000792  FC 42 D 098 7 02B         MP    WKSP.1+32(5),WORK_
                                        AREA.WA_RATE(3)
000798  D2 03 D 0F7 D 099         MVC   CURR_PAY(4),WKSP.1
                                        +33
```

In the listing shown in Figure 4-19, statement 91 shows that the data exception originates from the program trying to multiply WA_HOURS by WA_RATE. WA_HOURS and WA_RATE are defined in the structure WORK_AREA, which is a based variable, based on field WA_PTR. The value in the field WA_PTR is the address of the structure.

For our next example, we need to note that field WA_PTR is at offset CC within the current DSA (Figure 4-20). The current DSA is always pointed to by register 13.

Figure 4-20. Variable Storage Map Showing WA_PTR

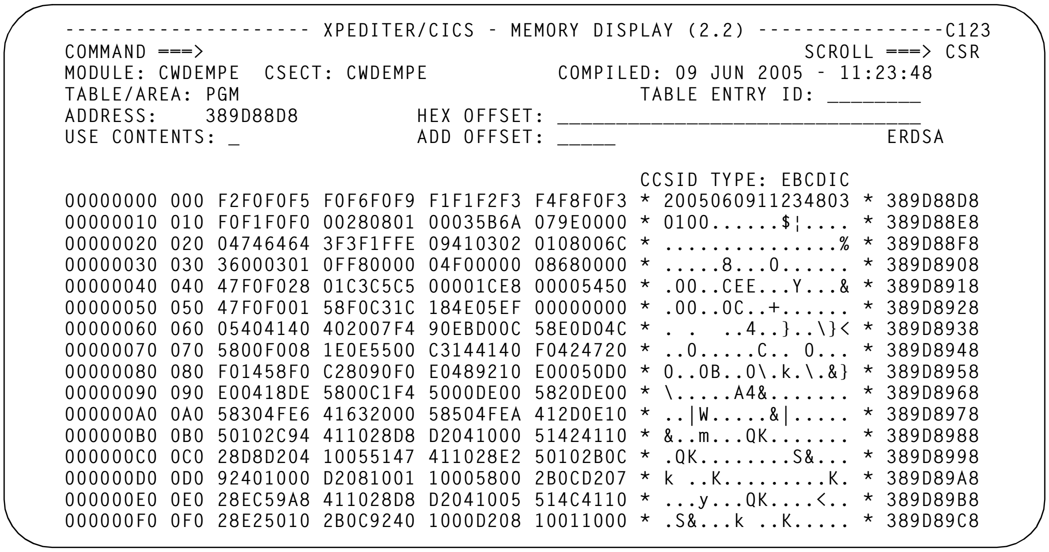
```
                                  VARIABLE STORAGE MAP
IDENTIFIER                        LEVEL         OFFSET       (HEX)   CLASS
.
.
.
EMP_TOTPAY                            1           2608         A30   AUTO
FILLER                                1           2615         A37   AUTO
WA_PTR                                1            204          CC   AUTO
PAYROLL_DATA_EMP999                   1           2665         A69   AUTO
PAY999_TYPE                           1           2665         A69   AUTO
.
.
.
```

To find the field containing the invalid data, you can view the structure on the Memory Display screen (2.2) by locating the address held in field WA_PTR.

Modifying Storage

1. Type **=2.2** in the COMMAND field and press Enter, or press PF14, to transfer to the Memory Display screen (2.2) as shown in Figure 4-21.

Figure 4-21. Viewing Program Storage on the Memory Display Screen (2.2)

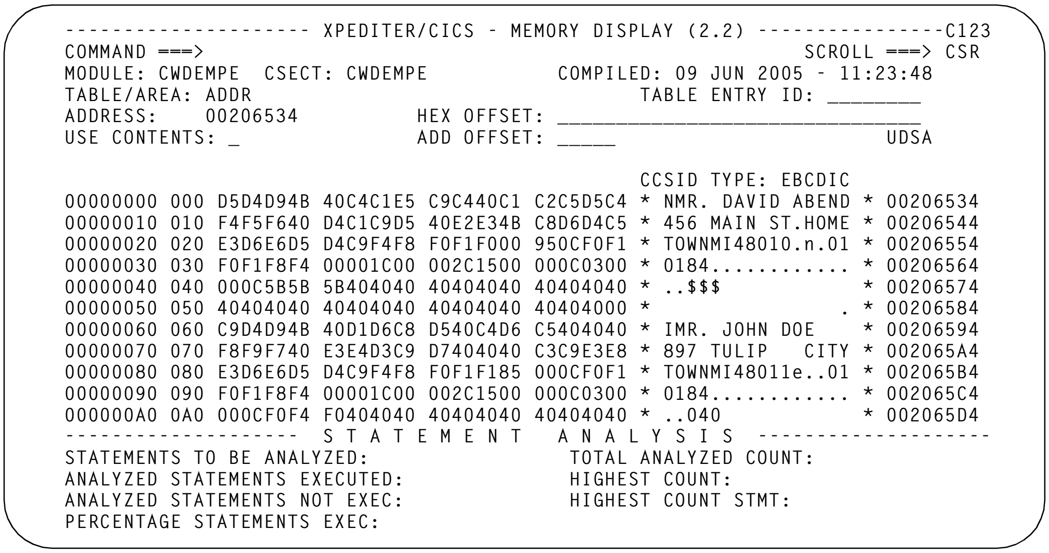


```
-------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
TABLE/AREA: PGM                                  TABLE ENTRY ID: ________
ADDRESS:    389D88D8          HEX OFFSET: _______________________________
USE CONTENTS: _               ADD OFFSET: _____                      ERDSA

                                                 CCSID TYPE: EBCDIC
00000000 000 F2F0F0F5 F0F6F0F9 F1F1F2F3 F4F8F0F3 * 2005060911234803 * 389D88D8
00000010 010 F0F1F0F0 00280801 00035B6A 079E0000 * 0100......$¦.... * 389D88E8
00000020 020 04746464 3F3F1FFE 09410302 0108006C * ...............% * 389D88F8
00000030 030 36000301 0FF80000 04F00000 08680000 * .....8...0...... * 389D8908
00000040 040 47F0F028 01C3C5C5 00001CE8 00005450 * .00..CEE...Y...& * 389D8918
00000050 050 47F0F001 58F0C31C 184E05EF 00000000 * .00..0C..+...... * 389D8928
00000060 060 05404140 402007F4 90EBD00C 58E0D04C * . .  ..4..}..\}< * 389D8938
00000070 070 5800F008 1E0E5500 C3144140 F0424720 * ..0.....C.. 0... * 389D8948
00000080 080 F01458F0 C28090F0 E0489210 E00050D0 * 0..0B..0\.k.\.&} * 389D8958
00000090 090 E00418DE 5800C1F4 5000DE00 5820DE00 * \.....A4&....... * 389D8968
000000A0 0A0 58304FE6 41632000 58504FEA 412D0E10 * ..|W.....&|..... * 389D8978
000000B0 0B0 50102C94 411028D8 D2041000 51424110 * &..m...QK....... * 389D8988
000000C0 0C0 28D8D204 10055147 411028E2 50102B0C * .QK........S&... * 389D8998
000000D0 0D0 92401000 D2081001 10005800 2B0CD207 * k ..K.........K. * 389D89A8
000000E0 0E0 28EC59A8 411028D8 D2041005 514C4110 * ...y...QK....<.. * 389D89B8
000000F0 0F0 28E25010 2B0C9240 1000D208 10011000 * .S&...k ..K..... * 389D89C8
```

The Memory Display screen (2.2) shows a hexadecimal dump of your test program. Any instruction or data in your program is available from here. Simply type the displacement you wish to see in the HEX (or ADD) OFFSET field.

2. To display storage for the work area containing the bad data (WA_PTR), overtype PGM in the TABLE/AREA field with **R13**, type **+CC** in the HEX OFFSET field, type **X** in the USE CONTENTS field, and press Enter. This positions the display to the address contained in field WA_PTR. Figure 4-22 shows the storage for the work area displayed on the Memory Display screen (2.2).

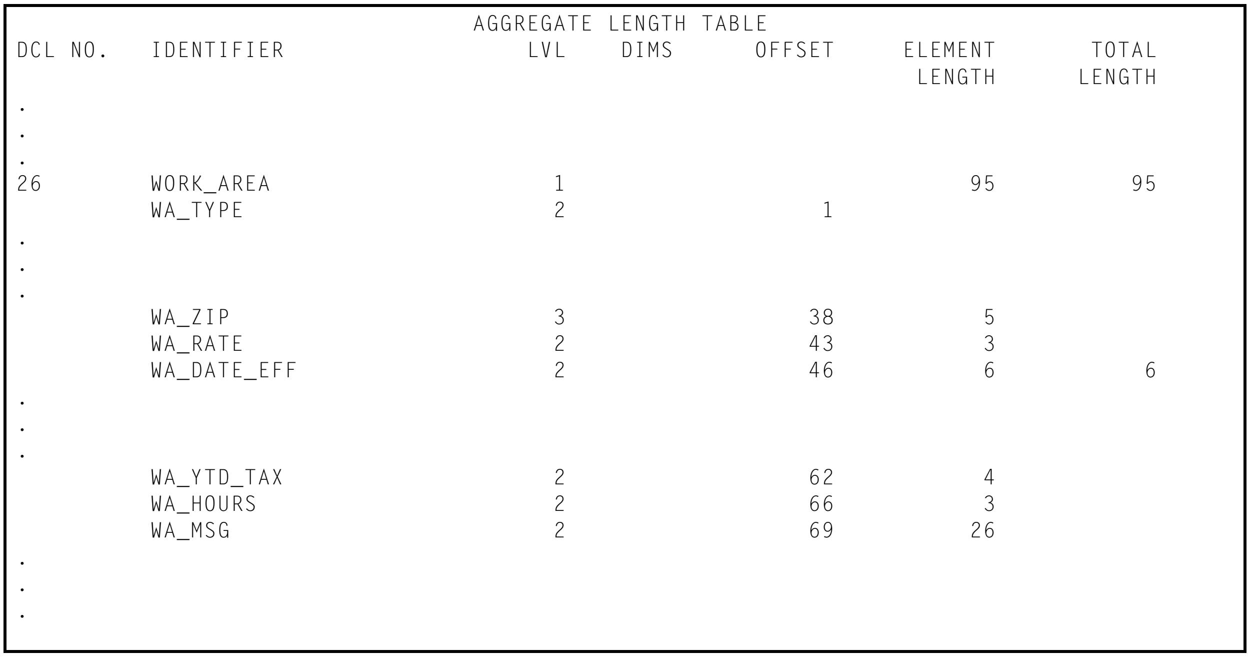
Figure 4-22. Displaying the Address of WA_PTR



```
-------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
TABLE/AREA: ADDR                                 TABLE ENTRY ID: ________
ADDRESS:    00206534          HEX OFFSET: _______________________________
USE CONTENTS: _               ADD OFFSET: _____                      UDSA

                                                 CCSID TYPE: EBCDIC
00000000 000 D5D4D94B 40C4C1E5 C9C440C1 C2C5D5C4 * NMR. DAVID ABEND * 00206534
00000010 010 F4F5F640 D4C1C9D5 40E2E34B C8D6D4C5 * 456 MAIN ST.HOME * 00206544
00000020 020 E3D6E6D5 D4C9F4F8 F0F1F000 950CF0F1 * TOWNMI48010.n.01 * 00206554
00000030 030 F0F1F8F4 00001C00 002C1500 000C0300 * 0184............ * 00206564
00000040 040 000C5B5B 5B404040 40404040 40404040 * ..$$$            * 00206574
00000050 050 40404040 40404040 40404040 40404000 *                . * 00206584
00000060 060 C9D4D94B 40D1D6C8 D540C4D6 C5404040 * IMR. JOHN DOE    * 00206594
00000070 070 F8F9F740 E3E4D3C9 D7404040 C3C9E3E8 * 897 TULIP   CITY * 002065A4
00000080 080 E3D6E6D5 D4C9F4F8 F0F1F185 000CF0F1 * TOWNMI48011e..01 * 002065B4
00000090 090 F0F1F8F4 00001C00 002C1500 000C0300 * 0184............ * 002065C4
000000A0 0A0 000CF0F4 F0404040 40404040 40404040 * ..040            * 002065D4
-------------------- S T A T E M E N T   A N A L Y S I S -------------------
STATEMENTS TO BE ANALYZED:                 TOTAL ANALYZED COUNT:
ANALYZED STATEMENTS EXECUTED:              HIGHEST COUNT:
ANALYZED STATEMENTS NOT EXEC:              HIGHEST COUNT STMT:
PERCENTAGE STATEMENTS EXEC:
```

3. From the Aggregate Length Table (condensed from the actual listing) shown in Figure 4-23, find the WORK_AREA structure. The structure shows the offsets and lengths of WA_RATE (3 bytes, offset 43), and WA_HOURS (3 bytes, offset 66).

Figure 4-23. Aggregate Length Table



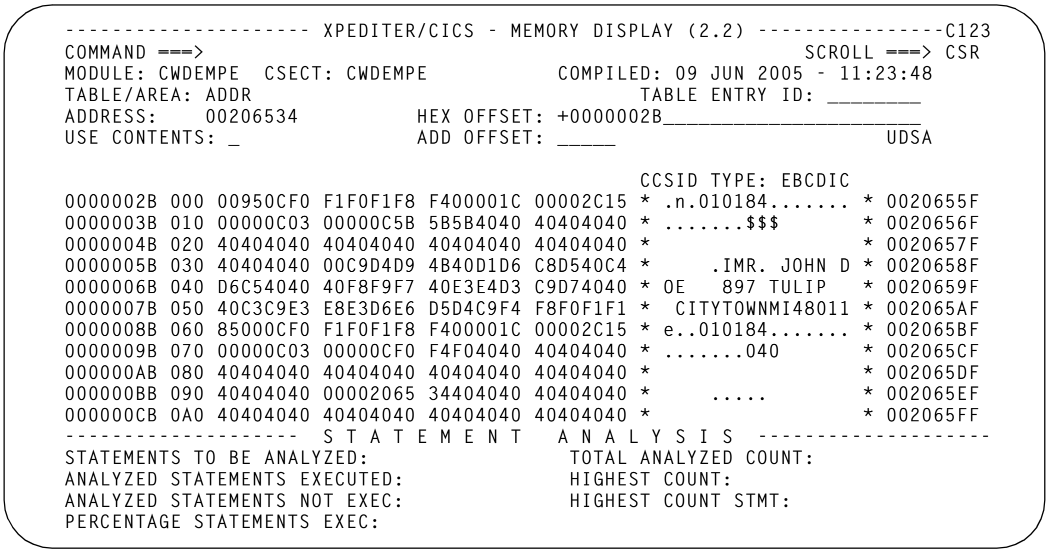
```
                                  AGGREGATE LENGTH TABLE
DCL NO.    IDENTIFIER                 LVL    DIMS      OFFSET     ELEMENT       TOTAL
                                                                   LENGTH      LENGTH
.
.
.
26         WORK_AREA                   1                               95          95
           WA_TYPE                     2                    1
.
.
.
           WA_ZIP                      3                   38           5
           WA_RATE                     2                   43           3
           WA_DATE_EFF                 2                   46           6           6
.
.
.
           WA_YTD_TAX                  2                   62           4
           WA_HOURS                    2                   66           3
           WA_MSG                      2                   69          26
.
.
.
```

Notice that the offsets are shown in decimal format. To convert them to hex format, use the CALC command as follows:

- On the COMMAND line, type **?? 43** and press Enter to convert the offset for WA_RATE. The result is 2B in hex format.
- To find the hex offset of WA_HOURS, type **?? 66** and press Enter. The result is 42.

4. To display the contents of your first multiplier (WA_RATE), tell Xpediter/CICS where to look by typing **+2B** in the HEX OFFSET field. When you press Enter, the display is positioned at the first data item as shown in Figure 4-24.

Figure 4-24. Viewing Data Fields on the Memory Display Screen (2.2)

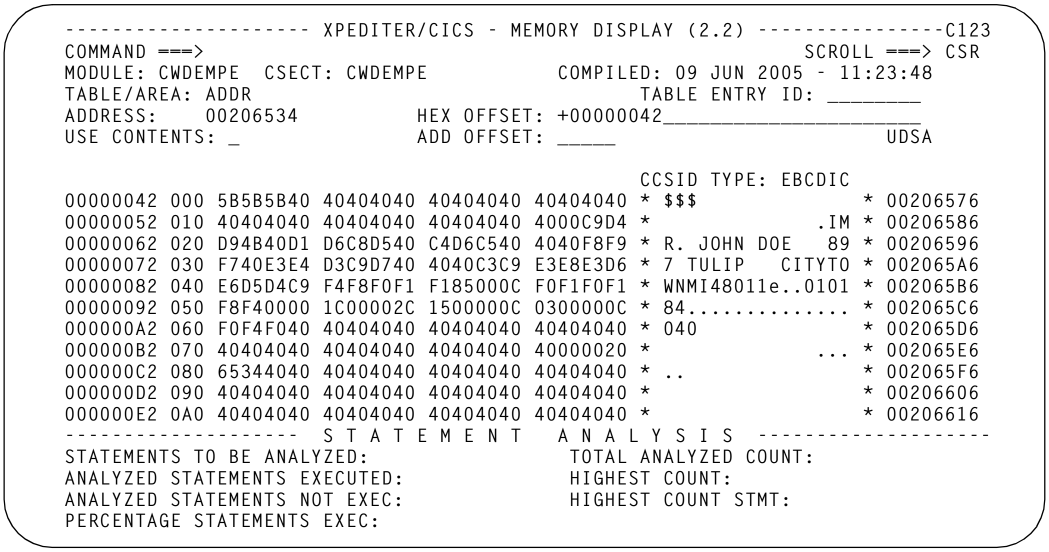


```
--------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48
TABLE/AREA: ADDR                                TABLE ENTRY ID: ________
ADDRESS:    00206534          HEX OFFSET: +0000002B______________________
USE CONTENTS: _               ADD OFFSET: _____                         UDSA

                                                CCSID TYPE: EBCDIC
0000002B 000 00950CF0 F1F0F1F8 F400001C 00002C15 * .n.010184....... * 0020655F
0000003B 010 00000C03 00000C5B 5B5B4040 40404040 * .......$$$       * 0020656F
0000004B 020 40404040 40404040 40404040 40404040 *                  * 0020657F
0000005B 030 40404040 00C9D4D9 4B40D1D6 C8D540C4 *     .IMR. JOHN D * 0020658F
0000006B 040 D6C54040 40F8F9F7 40E3E4D3 C9D74040 * OE   897 TULIP   * 0020659F
0000007B 050 40C3C9E3 E8E3D6E6 D5D4C9F4 F8F0F1F1 *  CITYTOWNMI48011 * 002065AF
0000008B 060 85000CF0 F1F0F1F8 F400001C 00002C15 * e..010184....... * 002065BF
0000009B 070 00000C03 00000CF0 F4F04040 40404040 * .......040       * 002065CF
000000AB 080 40404040 40404040 40404040 40404040 *                  * 002065DF
000000BB 090 40404040 00002065 34404040 40404040 *     .....        * 002065EF
000000CB 0A0 40404040 40404040 40404040 40404040 *                  * 002065FF
-------------------  S T A T E M E N T   A N A L Y S I S  -------------------
STATEMENTS TO BE ANALYZED:                 TOTAL ANALYZED COUNT:
ANALYZED STATEMENTS EXECUTED:              HIGHEST COUNT:
ANALYZED STATEMENTS NOT EXEC:              HIGHEST COUNT STMT:
PERCENTAGE STATEMENTS EXEC:
```

Note that the displayed data area contains valid packed information.

5. Display the displacement of the second data item (WA_HOURS) by typing **+42** in the cleared HEX OFFSET field. When you press Enter, the screen is positioned at the second data item as shown in Figure 4-25.

Figure 4-25. Viewing the Second Offset on the Memory Display Screen (2.2)



```
---------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
TABLE/AREA: ADDR                                 TABLE ENTRY ID: ________
ADDRESS:    00206534          HEX OFFSET: +00000042______________________
USE CONTENTS: _               ADD OFFSET: _____                      UDSA

                                                 CCSID TYPE: EBCDIC
00000042 000 5B5B5B40 40404040 40404040 40404040 * $$$              * 00206576
00000052 010 40404040 40404040 40404040 4000C9D4 *              .IM * 00206586
00000062 020 D94B40D1 D6C8D540 C4D6C540 4040F8F9 * R. JOHN DOE   89 * 00206596
00000072 030 F740E3E4 D3C9D740 4040C3C9 E3E8E3D6 * 7 TULIP   CITYTO * 002065A6
00000082 040 E6D5D4C9 F4F8F0F1 F185000C F0F1F0F1 * WNMI48011e..0101 * 002065B6
00000092 050 F8F40000 1C00002C 1500000C 0300000C * 84.............. * 002065C6
000000A2 060 F0F4F040 40404040 40404040 40404040 * 040              * 002065D6
000000B2 070 40404040 40404040 40404040 40000020 *              ... * 002065E6
000000C2 080 65344040 40404040 40404040 40404040 * ..               * 002065F6
000000D2 090 40404040 40404040 40404040 40404040 *                  * 00206606
000000E2 0A0 40404040 40404040 40404040 40404040 *                  * 00206616
-------------------  S T A T E M E N T   A N A L Y S I S  -------------------
STATEMENTS TO BE ANALYZED:                  TOTAL ANALYZED COUNT:
ANALYZED STATEMENTS EXECUTED:               HIGHEST COUNT:
ANALYZED STATEMENTS NOT EXEC:               HIGHEST COUNT STMT:
PERCENTAGE STATEMENTS EXEC:
```

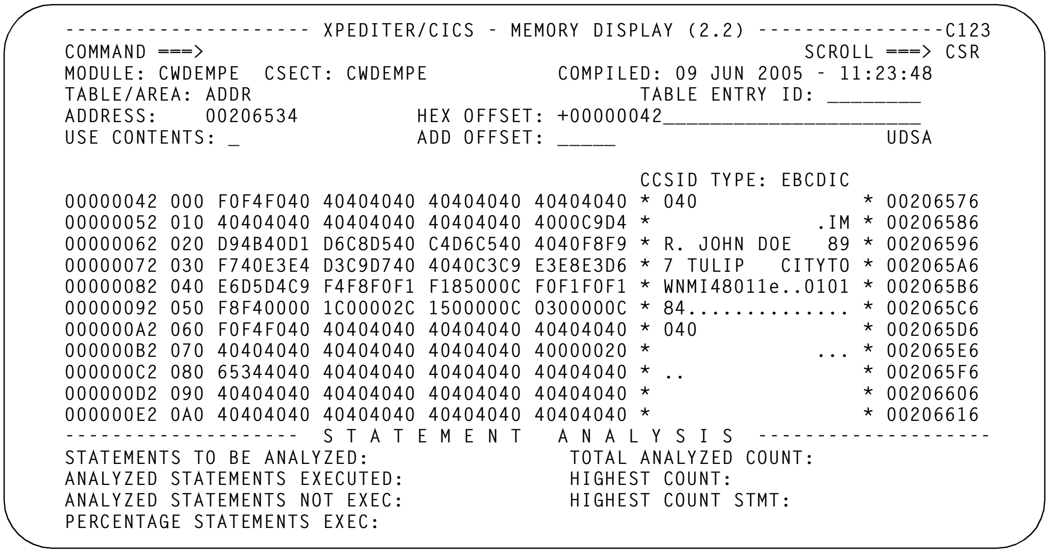
This time, the displayed field contains bad data. The three-byte field contains 5B5B5B, otherwise known as $$$. This error caused the ASRA.

6. Position the cursor on the bad data (**$$$**).

7. Replace the bad data by typing a valid number, such as **040**, over the data.

8. Press Enter to update the change. The area will contain valid data as shown in Figure 4-26.

Figure 4-26. Modifying Data on the Memory Display Screen (2.2)



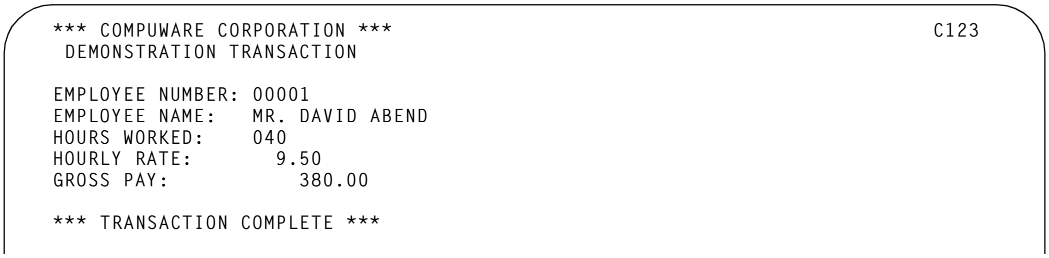
```
---------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
TABLE/AREA: ADDR                                 TABLE ENTRY ID: ________
ADDRESS:    00206534          HEX OFFSET: +00000042______________________
USE CONTENTS: _               ADD OFFSET: _____                      UDSA

                                                 CCSID TYPE: EBCDIC
00000042 000 F0F4F040 40404040 40404040 40404040 * 040              * 00206576
00000052 010 40404040 40404040 40404040 4000C9D4 *              .IM * 00206586
00000062 020 D94B40D1 D6C8D540 C4D6C540 4040F8F9 * R. JOHN DOE   89 * 00206596
00000072 030 F740E3E4 D3C9D740 4040C3C9 E3E8E3D6 * 7 TULIP   CITYTO * 002065A6
00000082 040 E6D5D4C9 F4F8F0F1 F185000C F0F1F0F1 * WNMI48011e..0101 * 002065B6
00000092 050 F8F40000 1C00002C 1500000C 0300000C * 84.............. * 002065C6
000000A2 060 F0F4F040 40404040 40404040 40404040 * 040              * 002065D6
000000B2 070 40404040 40404040 40404040 40000020 *              ... * 002065E6
000000C2 080 65344040 40404040 40404040 40404040 * ..               * 002065F6
000000D2 090 40404040 40404040 40404040 40404040 *                  * 00206606
000000E2 0A0 40404040 40404040 40404040 40404040 *                  * 00206616
-------------------  S T A T E M E N T   A N A L Y S I S  -------------------
STATEMENTS TO BE ANALYZED:                  TOTAL ANALYZED COUNT:
ANALYZED STATEMENTS EXECUTED:               HIGHEST COUNT:
ANALYZED STATEMENTS NOT EXEC:               HIGHEST COUNT STMT:
PERCENTAGE STATEMENTS EXEC:
```

Now that you have corrected the data causing the error, you can continue with the test.

9. Press PF22, or type **=2.20** and press Enter, to return to the Assembler Break/Abend screen (2.20).
10. Replace the value in the RESUME OFFSET field with the offset of the first instruction in statement 91 where the abend occurred — in this case, **77A**.
11. To continue your test, re-execute statement 91 by pressing PF12 or by typing **GO** in the COMMAND field and pressing Enter. This time, the transaction should not abend.

    The program finishes without further abends, concluding our exercise in sourceless debugging (Figure 4-27).

Figure 4-27. Transaction Complete on the Demonstration Transaction Screen



```
*** COMPUWARE CORPORATION ***                                              C123
 DEMONSTRATION TRANSACTION

EMPLOYEE NUMBER: 00001
EMPLOYEE NAME:   MR. DAVID ABEND
HOURS WORKED:    040
HOURLY RATE:       9.50
GROSS PAY:           380.00

*** TRANSACTION COMPLETE ***
```

Chapter 5. Debugging Subroutines

This chapter discusses how to test load modules that consist of multiple programs or control sections (CSECTs). The sections of the chapter show how to access source code and set breakpoints in calling and called programs, execute a program with multiple CSECTs, and return to the calling program. It also provides ways to select and exclude CSECTs.

Many modules consist of individual source members that are compiled or assembled separately, then link-edited together to create a load module. Xpediter/CICS lets you debug these CSECTs at the source level, even when the CSECT is not defined in your program resources.

The automatic CSECT support facility provides screens that list the CSECTs associated with each program. You choose the program for which you need to see the source. Facilities are also available so that IBM or vendor-supplied CSECTs may be excluded from automatic CSECT processing.

In this chapter, you will set a breakpoint in the calling program CWDEMPE, then access the source code for CWCDSUBA and set a breakpoint there. CWCDSUBA is a subroutine linked into CWDEMPE that does not have an entry in the program resources to define it to CICS. After setting the breakpoint, you will execute the transaction that runs CWDEMPE and step through the source in CWCDSUBA.

Note: If you cannot obtain a source listing for CWCDSUBA, your site may not be licensed for Assembler support. When the **NO SOURCE AVAILABLE** message appears, press PF1 for source information. If your site is not licensed for Assembler support, you can still read through this chapter to understand the concepts presented.

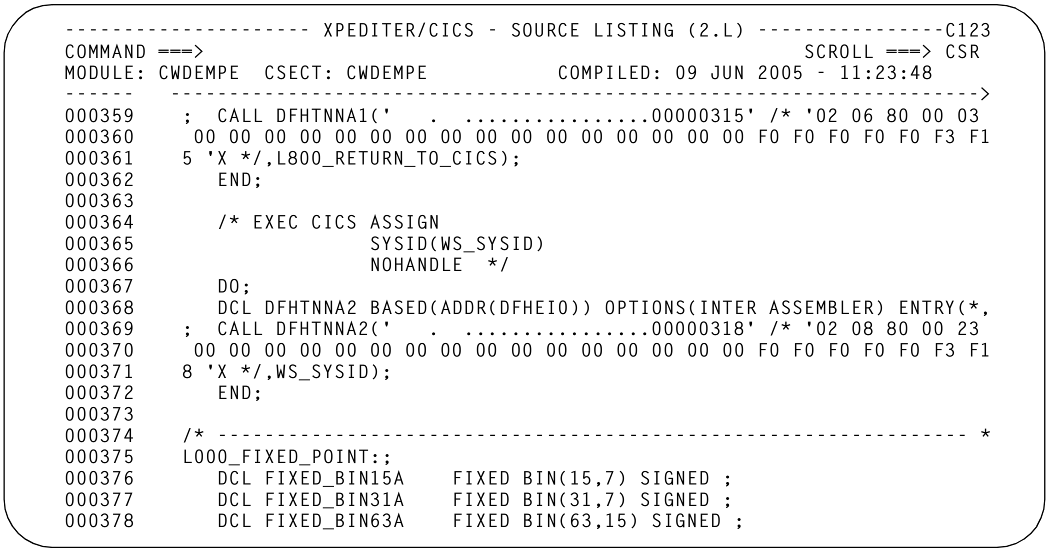
Note: The demonstrations in this chapter should be performed in Xpediter's standard operating mode — ***not*** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

Note: DLL modules are supported at the CSECT level on the 2.6.1 screen described later in this chapter. DLL modules are also supported at the function level via the 2.6.F screen. If you are debugging a DLL module containing multiple functions, refer to the *Xpediter/CICS Reference Manual* for further details.

Setting Breakpoints in a Calling Program

1. Type **XPED CWDEMPE** on a blank CICS screen. Press Enter to display source code for CWDEMPE on the Source Listing screen (2.L) as shown in Figure 5-1.

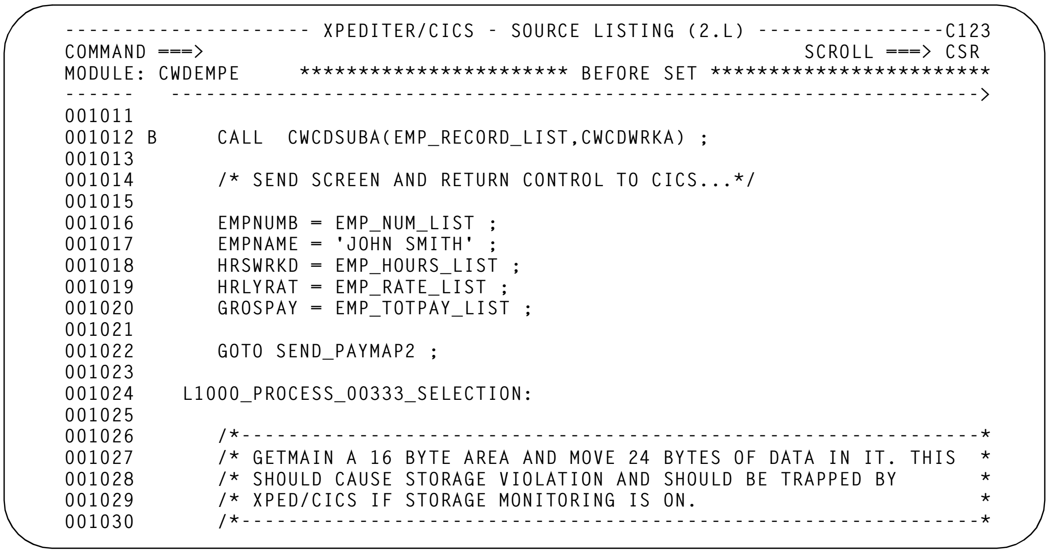
Figure 5-1. Source Listing Screen (2.L)



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48
------   -------------------------------------------------------------------->
000359    ;  CALL DFHTNNA1('   .  ................00000315' /* '02 06 80 00 03
000360     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0 F0 F0 F0 F0 F3 F1
000361    5 'X */,L800_RETURN_TO_CICS);
000362       END;
000363
000364       /* EXEC CICS ASSIGN
000365                    SYSID(WS_SYSID)
000366                    NOHANDLE  */
000367       DO;
000368       DCL DFHTNNA2 BASED(ADDR(DFHEIO)) OPTIONS(INTER ASSEMBLER) ENTRY(*,
000369    ;  CALL DFHTNNA2('   .  ................00000318' /* '02 08 80 00 23
000370     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0 F0 F0 F0 F0 F3 F1
000371    8 'X */,WS_SYSID);
000372       END;
000373
000374    /* ---------------------------------------------------------------- *
000375    L000_FIXED_POINT:;
000376       DCL FIXED_BIN15A    FIXED BIN(15,7) SIGNED ;
000377       DCL FIXED_BIN31A    FIXED BIN(31,7) SIGNED ;
000378       DCL FIXED_BIN63A    FIXED BIN(63,15) SIGNED ;
```

2. Type **F CWCDSUBA** in the COMMAND field. Press Enter to find the call to CWCDSUBA.
3. Type the **B** (Before) line command on the statement number for the call, which is statement 1012 in this example, and press Enter. The **B** flag on the Source Listing screen (2.L) shows that the breakpoint has been set (Figure 5-2).

Figure 5-2. Breakpoint Set on the Source Listing Screen (2.L)

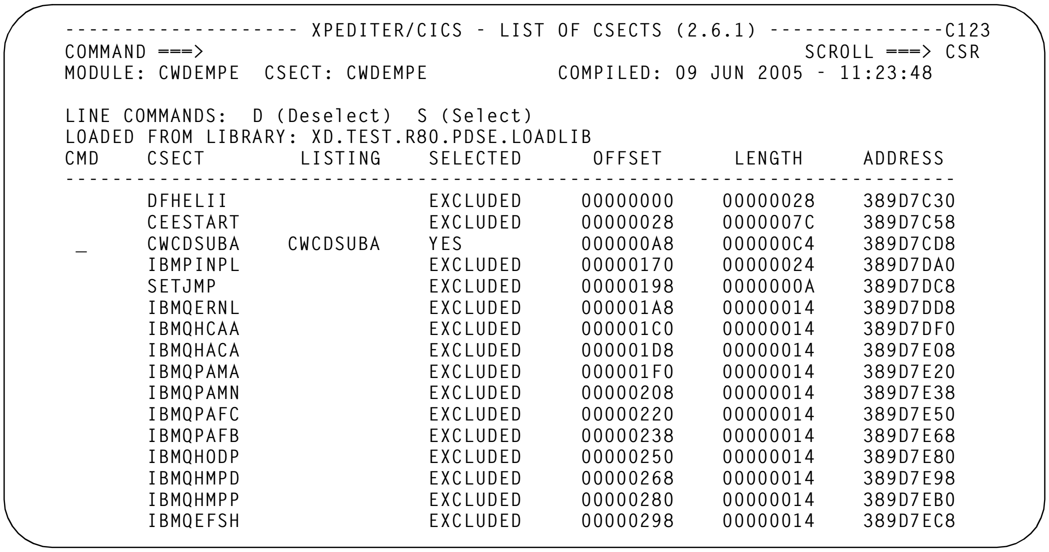


```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE     *********************** BEFORE SET ***********************
------   -------------------------------------------------------------------->
001011
001012 B     CALL  CWCDSUBA(EMP_RECORD_LIST,CWCDWRKA) ;
001013
001014       /* SEND SCREEN AND RETURN CONTROL TO CICS...*/
001015
001016       EMPNUMB = EMP_NUM_LIST ;
001017       EMPNAME = 'JOHN SMITH' ;
001018       HRSWRKD = EMP_HOURS_LIST ;
001019       HRLYRAT = EMP_RATE_LIST ;
001020       GROSPAY = EMP_TOTPAY_LIST ;
001021
001022       GOTO SEND_PAYMAP2 ;
001023
001024    L1000_PROCESS_00333_SELECTION:
001025
001026       /*---------------------------------------------------------------*
001027       /* GETMAIN A 16 BYTE AREA AND MOVE 24 BYTES OF DATA IN IT. THIS  *
001028       /* SHOULD CAUSE STORAGE VIOLATION AND SHOULD BE TRAPPED BY       *
001029       /* XPED/CICS IF STORAGE MONITORING IS ON.                        *
001030       /*---------------------------------------------------------------*
```

4. Type **=2.6.1** in the COMMAND field and press Enter to display the List of CSECTs screen (2.6.1) (Figure 5-3).

   This screen shows all the CSECTs linked together to form the load module CWDEMPE. The SELECTED field indicates whether source is shown for that CSECT. CICS (DFH) and PL/I (IBM) modules are automatically excluded. The source for CWDEMPE displays automatically — you do not need to select it.

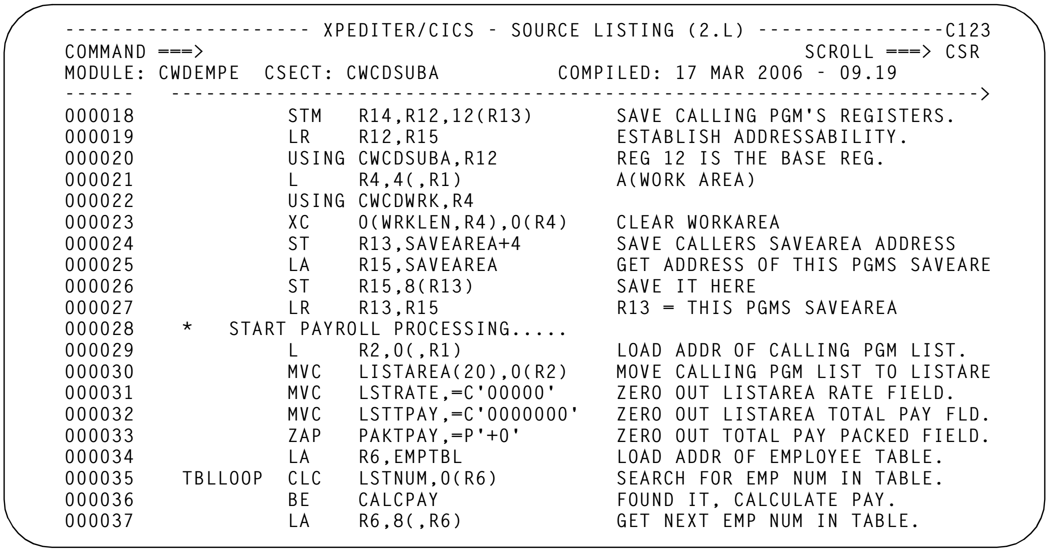
Figure 5-3. List of CSECTs Screen (2.6.1)



```
------------------- XPEDITER/CICS - LIST OF CSECTS (2.6.1) --------------C123
COMMAND ===>                                                 SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48

LINE COMMANDS:  D (Deselect)  S (Select)
LOADED FROM LIBRARY: XD.TEST.R80.PDSE.LOADLIB
CMD    CSECT        LISTING    SELECTED      OFFSET      LENGTH     ADDRESS
---------------------------------------------------------------------------
       DFHELII                 EXCLUDED     00000000    00000028    389D7C30
       CEESTART                EXCLUDED     00000028    0000007C    389D7C58
 _     CWCDSUBA     CWCDSUBA   YES          000000A8    000000C4    389D7CD8
       IBMPINPL                EXCLUDED     00000170    00000024    389D7DA0
       SETJMP                  EXCLUDED     00000198    0000000A    389D7DC8
       IBMQERNL                EXCLUDED     000001A8    00000014    389D7DD8
       IBMQHCAA                EXCLUDED     000001C0    00000014    389D7DF0
       IBMQHACA                EXCLUDED     000001D8    00000014    389D7E08
       IBMQPAMA                EXCLUDED     000001F0    00000014    389D7E20
       IBMQPAMN                EXCLUDED     00000208    00000014    389D7E38
       IBMQPAFC                EXCLUDED     00000220    00000014    389D7E50
       IBMQPAFB                EXCLUDED     00000238    00000014    389D7E68
       IBMQHODP                EXCLUDED     00000250    00000014    389D7E80
       IBMQHMPD                EXCLUDED     00000268    00000014    389D7E98
       IBMQHMPP                EXCLUDED     00000280    00000014    389D7EB0
       IBMQEFSH                EXCLUDED     00000298    00000014    389D7EC8
```

5. Type **S** next to CWCDSUBA in the CMD field and press Enter.
6. Type **CWCDSUBA** in the CSECT field.
7. Type **=2.L** in the COMMAND field and press Enter. The Source Listing screen (2.L) is displayed showing source for CWCDSUBA (Figure 5-4).

Figure 5-4. Source Listing Screen (2.L) Showing CWCDSUBA

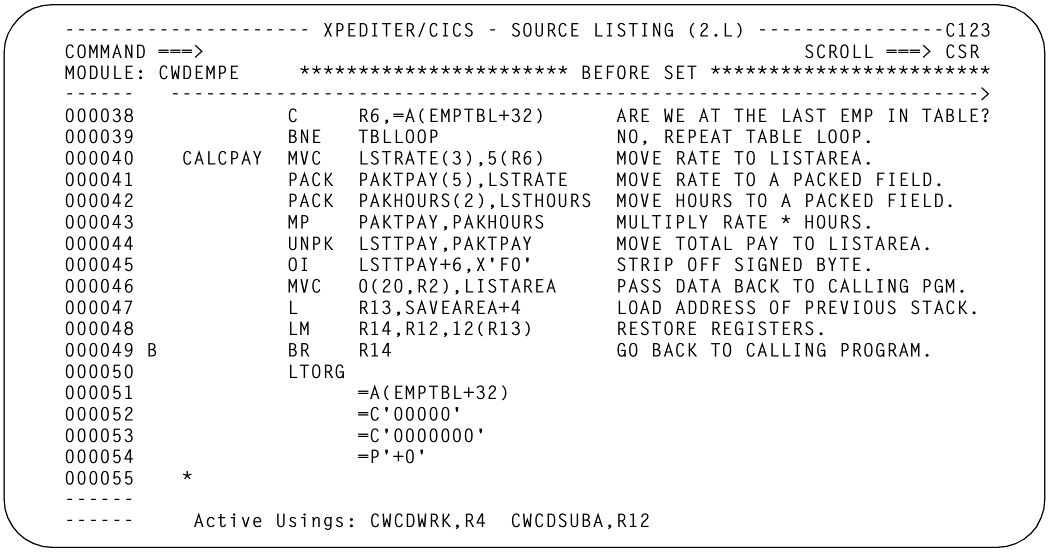


```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ---------------C123
COMMAND ===>                                                 SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWCDSUBA         COMPILED: 17 MAR 2006 - 09.19
------   ------------------------------------------------------------------->
000018            STM   R14,R12,12(R13)       SAVE CALLING PGM'S REGISTERS.
000019            LR    R12,R15               ESTABLISH ADDRESSABILITY.
000020            USING CWCDSUBA,R12          REG 12 IS THE BASE REG.
000021            L     R4,4(,R1)             A(WORK AREA)
000022            USING CWCDWRK,R4
000023            XC    0(WRKLEN,R4),0(R4)    CLEAR WORKAREA
000024            ST    R13,SAVEAREA+4        SAVE CALLERS SAVEAREA ADDRESS
000025            LA    R15,SAVEAREA          GET ADDRESS OF THIS PGMS SAVEARE
000026            ST    R15,8(R13)            SAVE IT HERE
000027            LR    R13,R15               R13 = THIS PGMS SAVEAREA
000028     *   START PAYROLL PROCESSING.....
000029            L     R2,0(,R1)             LOAD ADDR OF CALLING PGM LIST.
000030            MVC   LISTAREA(20),0(R2)    MOVE CALLING PGM LIST TO LISTARE
000031            MVC   LSTRATE,=C'00000'     ZERO OUT LISTAREA RATE FIELD.
000032            MVC   LSTTPAY,=C'0000000'   ZERO OUT LISTAREA TOTAL PAY FLD.
000033            ZAP   PAKTPAY,=P'+0'        ZERO OUT TOTAL PAY PACKED FIELD.
000034            LA    R6,EMPTBL             LOAD ADDR OF EMPLOYEE TABLE.
000035     TBLLOOP CLC  LSTNUM,0(R6)          SEARCH FOR EMP NUM IN TABLE.
000036            BE    CALCPAY               FOUND IT, CALCULATE PAY.
000037            LA    R6,8(,R6)             GET NEXT EMP NUM IN TABLE.
```

8. Press PF8 to scroll down through the listing.
9. Type the **B** (Before) line command on the statement number for the instruction BR R14 (statement 49 in this example) and press Enter. This sets a breakpoint at the branch back to the main program (Figure 5-5).

   You have set a breakpoint in the main program at the call to the subroutine and a breakpoint in the subroutine just before the return to the main program. The programs are ready to test.

Figure 5-5. Breakpoint Set on the Source Listing Screen (2.L)

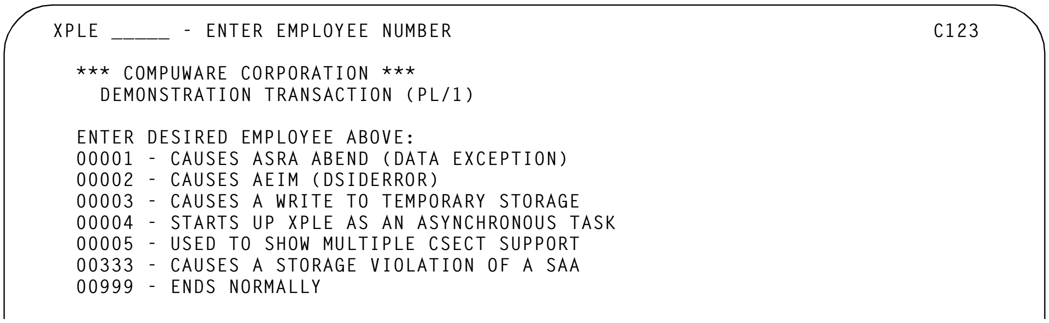


```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ---------------C123
COMMAND ===>                                                     SCROLL ===> CSR
MODULE: CWDEMPE     *********************** BEFORE SET ***********************
------   ------------------------------------------------------------------->
000038              C     R6,=A(EMPTBL+32)      ARE WE AT THE LAST EMP IN TABLE?
000039              BNE   TBLLOOP               NO, REPEAT TABLE LOOP.
000040    CALCPAY   MVC   LSTRATE(3),5(R6)      MOVE RATE TO LISTAREA.
000041              PACK  PAKTPAY(5),LSTRATE    MOVE RATE TO A PACKED FIELD.
000042              PACK  PAKHOURS(2),LSTHOURS  MOVE HOURS TO A PACKED FIELD.
000043              MP    PAKTPAY,PAKHOURS      MULTIPLY RATE * HOURS.
000044              UNPK  LSTTPAY,PAKTPAY       MOVE TOTAL PAY TO LISTAREA.
000045              OI    LSTTPAY+6,X'F0'       STRIP OFF SIGNED BYTE.
000046              MVC   0(20,R2),LISTAREA     PASS DATA BACK TO CALLING PGM.
000047              L     R13,SAVEAREA+4        LOAD ADDRESS OF PREVIOUS STACK.
000048              LM    R14,R12,12(R13)       RESTORE REGISTERS.
000049 B            BR    R14                   GO BACK TO CALLING PROGRAM.
000050              LTORG
000051                    =A(EMPTBL+32)
000052                    =C'00000'
000053                    =C'0000000'
000054                    =P'+0'
000055    *
------
------     Active Usings: CWCDWRK,R4  CWCDSUBA,R12
```

Executing a Program Containing Multiple CSECTs

1. Press Clear to return to CICS and run the program.
2. Type **XPLE** in the upper left corner of the screen and press Enter to display the Demonstration Transaction screen (Figure 5-6).

Figure 5-6. Demonstration Transaction Screen

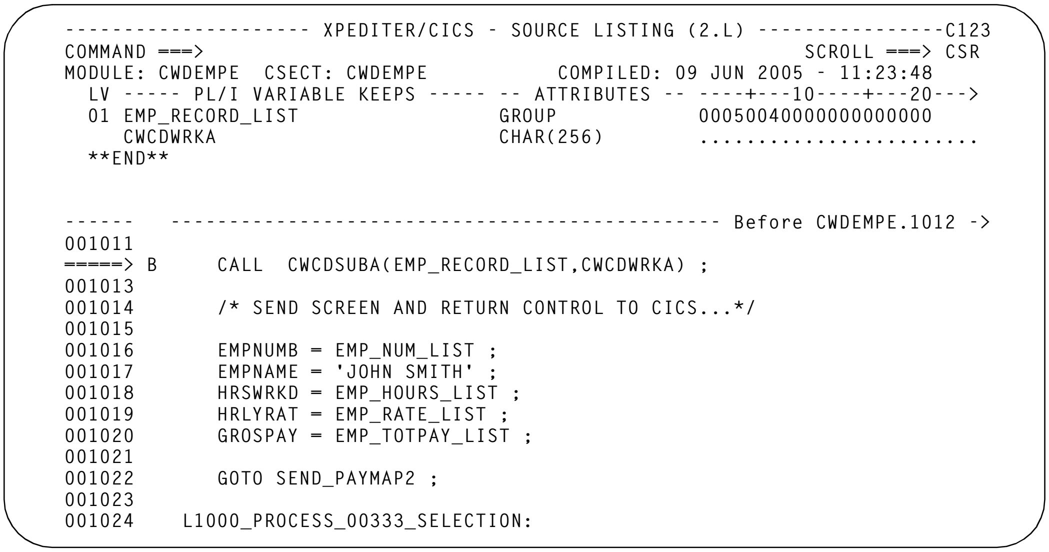


```
XPLE _____ - ENTER EMPLOYEE NUMBER                                       C123

  *** COMPUWARE CORPORATION ***
    DEMONSTRATION TRANSACTION (PL/1)

  ENTER DESIRED EMPLOYEE ABOVE:
  00001 - CAUSES ASRA ABEND (DATA EXCEPTION)
  00002 - CAUSES AEIM (DSIDERROR)
  00003 - CAUSES A WRITE TO TEMPORARY STORAGE
  00004 - STARTS UP XPLE AS AN ASYNCHRONOUS TASK
  00005 - USED TO SHOW MULTIPLE CSECT SUPPORT
  00333 - CAUSES A STORAGE VIOLATION OF A SAA
  00999 - ENDS NORMALLY
```

3. Type **00005** in the field preceding ENTER EMPLOYEE NUMBER and press Enter. The Source Listing screen (2.L) appears as shown in Figure 5-7.

   Note that the program is stopped before the call to the subroutine.

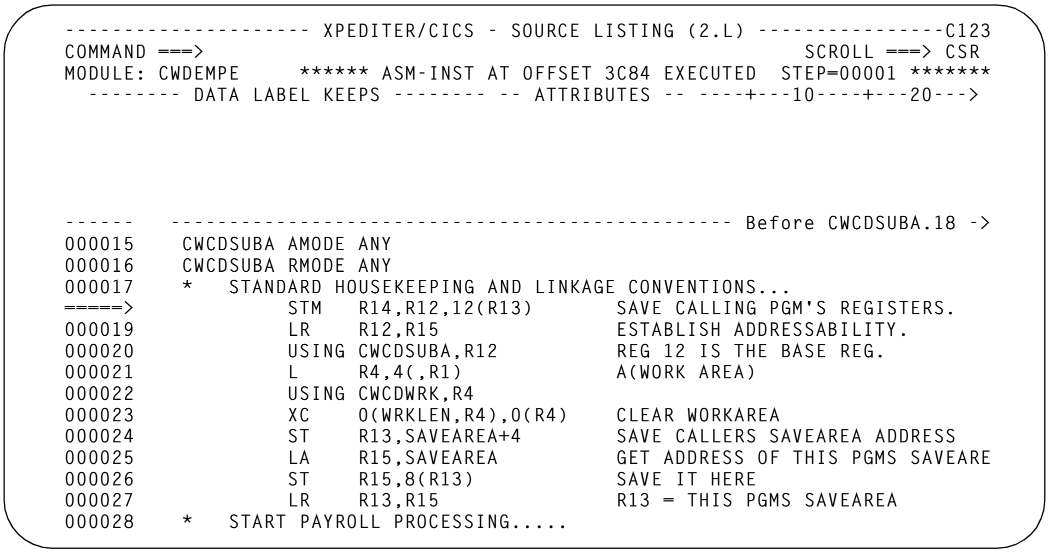
Figure 5-7. Source Listing Screen (2.L) Showing a Breakpoint Before the Subroutine



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
  LV ----- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
  01 EMP_RECORD_LIST                 GROUP            00050040000000000000
     CWCDWRKA                        CHAR(256)        ........................
  **END**


------   ---------------------------------------------- Before CWDEMPE.1012 ->
001011
=====> B     CALL  CWCDSUBA(EMP_RECORD_LIST,CWCDWRKA) ;
001013
001014       /* SEND SCREEN AND RETURN CONTROL TO CICS...*/
001015
001016       EMPNUMB = EMP_NUM_LIST ;
001017       EMPNAME = 'JOHN SMITH' ;
001018       HRSWRKD = EMP_HOURS_LIST ;
001019       HRLYRAT = EMP_RATE_LIST ;
001020       GROSPAY = EMP_TOTPAY_LIST ;
001021
001022       GOTO SEND_PAYMAP2 ;
001023
001024    L1000_PROCESS_00333_SELECTION:
```

4. Press PF9 (GO 1) to follow the logic into CWCDSUBA. The Source Listing screen (2.L) is displayed for CWCDSUBA (Figure 5-8).

Figure 5-8. Source Listing Screen (2.L) for CWCDSUBA

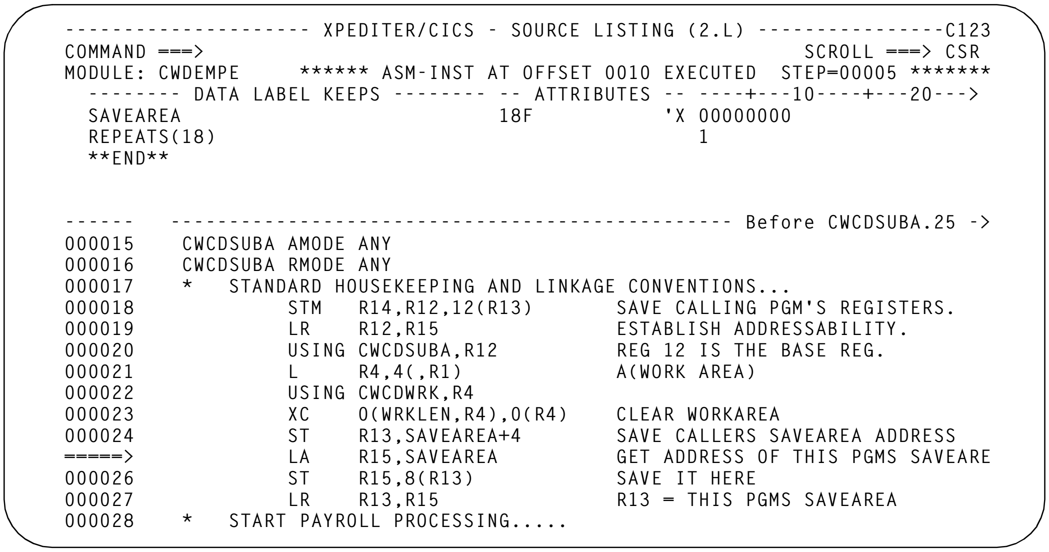


```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE     ****** ASM-INST AT OFFSET 3C84 EXECUTED  STEP=00001 *******
  -------- DATA LABEL KEEPS -------- -- ATTRIBUTES -- ----+---10----+---20--->




------   ----------------------------------------------- Before CWCDSUBA.18 ->
000015    CWCDSUBA AMODE ANY
000016    CWCDSUBA RMODE ANY
000017    *   STANDARD HOUSEKEEPING AND LINKAGE CONVENTIONS...
=====>             STM   R14,R12,12(R13)       SAVE CALLING PGM'S REGISTERS.
000019             LR    R12,R15               ESTABLISH ADDRESSABILITY.
000020             USING CWCDSUBA,R12          REG 12 IS THE BASE REG.
000021             L     R4,4(,R1)             A(WORK AREA)
000022             USING CWCDWRK,R4
000023             XC    0(WRKLEN,R4),0(R4)    CLEAR WORKAREA
000024             ST    R13,SAVEAREA+4        SAVE CALLERS SAVEAREA ADDRESS
000025             LA    R15,SAVEAREA          GET ADDRESS OF THIS PGMS SAVEARE
000026             ST    R15,8(R13)            SAVE IT HERE
000027             LR    R13,R15               R13 = THIS PGMS SAVEAREA
000028    *   START PAYROLL PROCESSING.....
```

5. Type **GO 5 1** in the COMMAND field and press Enter. Watch closely while Xpediter/CICS “slow steps” through CWCDSUBA (Figure 5-9). When five steps have executed, Xpediter/CICS displays the message

```
****** ASM-INST AT OFFSET 0010 EXECUTED  STEP=00005   *******
```

Figure 5-9. Executing GO 5 1 on the Source Listing Screen (2.L)

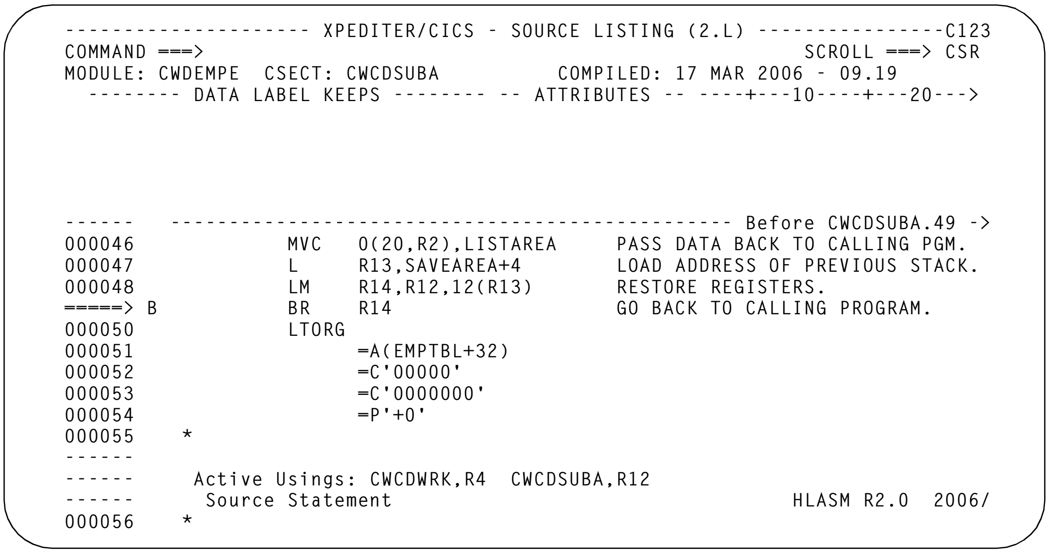


```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                     SCROLL ===> CSR
MODULE: CWDEMPE     ****** ASM-INST AT OFFSET 0010 EXECUTED  STEP=00005 *******
  -------- DATA LABEL KEEPS -------- -- ATTRIBUTES -- ----+---10----+---20--->
  SAVEAREA                         18F           'X 00000000
  REPEATS(18)                                       1
  **END**


------   ------------------------------------------------ Before CWCDSUBA.25 ->
000015    CWCDSUBA AMODE ANY
000016    CWCDSUBA RMODE ANY
000017    *   STANDARD HOUSEKEEPING AND LINKAGE CONVENTIONS...
000018             STM   R14,R12,12(R13)       SAVE CALLING PGM'S REGISTERS.
000019             LR    R12,R15               ESTABLISH ADDRESSABILITY.
000020             USING CWCDSUBA,R12          REG 12 IS THE BASE REG.
000021             L     R4,4(,R1)             A(WORK AREA)
000022             USING CWCDWRK,R4
000023             XC    0(WRKLEN,R4),0(R4)    CLEAR WORKAREA
000024             ST    R13,SAVEAREA+4        SAVE CALLERS SAVEAREA ADDRESS
=====>             LA    R15,SAVEAREA          GET ADDRESS OF THIS PGMS SAVEARE
000026             ST    R15,8(R13)            SAVE IT HERE
000027             LR    R13,R15               R13 = THIS PGMS SAVEAREA
000028    *   START PAYROLL PROCESSING.....
```

Returning to the Calling Program

1. Press PF12 (GO) to resume execution of the program. The Source Listing screen (2.L) is displayed showing the breakpoint at the branch instruction back to the calling program (Figure 5-10).

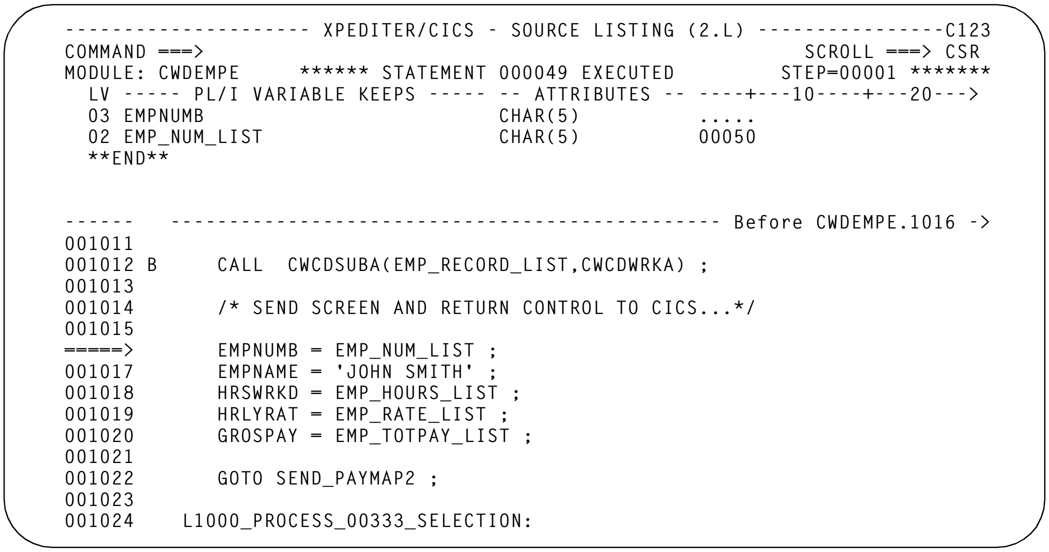
Figure 5-10. Source Listing Screen (2.L)



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                     SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWCDSUBA          COMPILED: 17 MAR 2006 - 09.19
  -------- DATA LABEL KEEPS -------- -- ATTRIBUTES -- ----+---10----+---20--->




------   ------------------------------------------------ Before CWCDSUBA.49 ->
000046             MVC   0(20,R2),LISTAREA     PASS DATA BACK TO CALLING PGM.
000047             L     R13,SAVEAREA+4        LOAD ADDRESS OF PREVIOUS STACK.
000048             LM    R14,R12,12(R13)       RESTORE REGISTERS.
=====> B           BR    R14                   GO BACK TO CALLING PROGRAM.
000050             LTORG
000051                   =A(EMPTBL+32)
000052                   =C'00000'
000053                   =C'0000000'
000054                   =P'+0'
000055    *
------
------     Active Usings: CWCDWRK,R4  CWCDSUBA,R12
------      Source Statement                                HLASM R2.0  2006/
000056    *
```

2. Press PF9 (GO 1) to step back to the calling program (Figure 5-11).

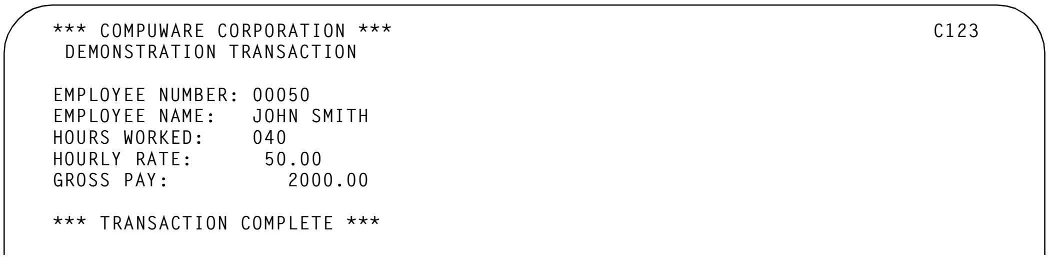
Figure 5-11. Source Listing Screen (2.L) of the Calling Program



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ---------------C123
COMMAND ===>                                                 SCROLL ===> CSR
MODULE: CWDEMPE     ****** STATEMENT 000049 EXECUTED        STEP=00001 *******
  LV ----- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
  03 EMPNUMB                         CHAR(5)          .....
  02 EMP_NUM_LIST                    CHAR(5)          00050
  **END**


------   ---------------------------------------------- Before CWDEMPE.1016 ->
001011
001012 B     CALL  CWCDSUBA(EMP_RECORD_LIST,CWCDWRKA) ;
001013
001014       /* SEND SCREEN AND RETURN CONTROL TO CICS...*/
001015
=====>       EMPNUMB = EMP_NUM_LIST ;
001017       EMPNAME = 'JOHN SMITH' ;
001018       HRSWRKD = EMP_HOURS_LIST ;
001019       HRLYRAT = EMP_RATE_LIST ;
001020       GROSPAY = EMP_TOTPAY_LIST ;
001021
001022       GOTO SEND_PAYMAP2 ;
001023
001024    L1000_PROCESS_00333_SELECTION:
```

3. Press PF12 to continue execution of the calling program. The screen displays the ***** TRANSACTION COMPLETE ***** message (Figure 5-12).

Figure 5-12. Demonstration Transaction Screen



```
*** COMPUWARE CORPORATION ***                                        C123
 DEMONSTRATION TRANSACTION

EMPLOYEE NUMBER: 00050
EMPLOYEE NAME:   JOHN SMITH
HOURS WORKED:    040
HOURLY RATE:      50.00
GROSS PAY:          2000.00

*** TRANSACTION COMPLETE ***
```

Stepping Through Subroutines and User Condition Handlers

This section explains how Xpediter's stepping function works with selected and unselected static and dynamic subroutines and Language Environment user condition handlers (UCHs).

Dynamically Called Subroutines

As demonstrated in the previous sections, Xpediter can step (GO*n*) through both dynamic and static subroutines. At execution time, support is essentially the same.

Support differs, however, at the time breakpoints are set. You can set a breakpoint in a static subroutine only after it has been selected, but Xpediter lets you set a breakpoint in a dynamic subroutine with or without first selecting it. Xpediter allows the breakpoint because dynamic subroutines are independent load modules which Xpediter cannot distinguish from main programs until execution time. A module might act as a dynamic subroutine under one transaction, then be invoked as the main program of another transaction.

Remember, however, that if a breakpoint is set in a dynamic subroutine without first selecting it, when a call to it from the main program is **stepped**, the breakpoint will be ignored. The same thing would happen if you set a breakpoint in a selected static subroutine, then deselected it. The breakpoint in the static subroutine would be ignored when the call to it was stepped.

Dynamic subroutines can be selected on the List of CSECTs screen (2.6.1) by first typing the module name in the MODULE field at the top of the screen. After Xpediter returns the CSECT information, the module can be selected in the same way as any other CSECT.

Automatic Selection of CSECTs

Under certain circumstances, Xpediter will automatically select a CSECT, generating an entry on the List of CSECTs screen (2.6.1). Auto-selection depends on a number of factors such as whether or not the program is being stepped and whether or not the load module and the main CSECT share the same name. Table 5-1 summarizes Xpediter's operation in various situations.

Table 5-1. Stepping, Resuming, and Auto-Selection of CSECTs

Operation	Xpediter Action
Module load via 2.L or 2.6.1	Xpediter will auto-select the main program CSECT if its name is different from the name of the load module.
Breakpoint or abend encountered at transaction startup	Xpediter will halt execution and auto-select the CSECT, if it is not the mainline program.
Breakpoint or abend after RESUME or GO	Xpediter will halt execution and auto-select the CSECT, if it is not the mainline program.
Breakpoint or abend when stepping (GO *n*)	• If the breakpoint or abend is encountered in the main program or a CSECT already selected on the 2.6.1 screen, Xpediter will halt execution. • Xpediter will ignore breakpoints in subroutines that are not selected. For example, if a call statement to an unselected dynamic subroutine is stepped, Xpediter will ignore any breakpoints in the subroutine and position to the statement after the call. • If abend occurs in unselected subroutine, Xpediter will position to the call statement in the calling module.
Selected CSECT in call sequence when stepping (GO *n*)	Xpediter will step into the selected CSECT and indicate that a program boundary has been crossed. For example, if stepping a mainline program call statement to an unselected subroutine which in turn calls a *selected* subroutine, Xpediter will step into the selected subroutine and indicate that a program boundary has been crossed. The intermediate unselected CSECT is executed transparently.

As shown in the table, with the GO (or RESUME) command, Xpediter will stop at a breakpoint regardless of the CSECT's selection status. If the CSECT is unselected and not the main program CSECT, Xpediter will auto-select it. The CSECT will remain selected until it is manually unselected or the session is ended.

The stepping function (GO *n*), however, stops at breakpoints only in selected CSECTs. If a CSECT is unselected, stepping will execute it transparently — regardless of whether or not it contains a breakpoint — and the CSECT will not be auto-selected.

Language Environment User Condition Handlers

Xpediter makes it possible for you to step through any user condition handler (UCH) registered with Language Environment's callable services. The way Xpediter stepping works with UCHs is similar to the stepping of subroutines. Xpediter's trace and storage protection monitoring functions treat UCHs as independent units, following the same rules as for EXEC CICS LINK operations. Refer to the *Xpediter/CICS Reference Manual* for more information.

UCHs are driven in the following situations:

- When the Language Environment condition manager detects a condition
- When an application issues a call signal
- When an abend occurs.

All three situations are treated as conditions. If a statement being stepped causes a condition to be raised, Xpediter will step into the UCH, but only if it is selected on the List of CSECTs screen (2.6.1).

Note that if the condition is the result of an abend and Xpediter was set up on the Trap Abend screen (1.6) to trap it, Xpediter's trap will preempt Language Environment's condition handler. To allow the UCH to get control, the TRAP ABEND option on the 1.6 screen should be set to NO. See the *Xpediter/CICS Reference Manual* for more information on the trap facility.

If the UCH is able to correct the condition and retry, Xpediter will step back into the main program or selected subroutine at the location where execution is being resumed.

Table 5-2 summarizes Xpediter's criteria and resulting actions in regard to UCHs.

Table 5-2. Stepping User Condition Handlers

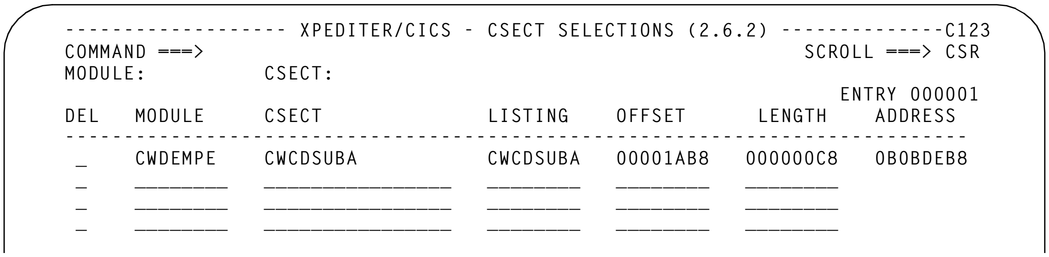
Situation When Stepping (GO *n*)	**Xpediter Action**
Statement raises a condition	• Xpediter will step into the UCH if it was selected on the 2.6.1 screen. • Unlike the stepping of CSECTs, if the UCH is not selected but a subroutine is, Xpediter will not step into the lower-level subroutine. For example, if a mainline program add statement causes an abend while stepping, and the UCH is not selected, Xpediter will ignore any subroutines called by the UCH even if they are selected. If the UCH is selected, Xpediter will step into it and any of its subroutines that are also selected.
UCH percolates or promotes condition.	When a statement is being stepped and causes a condition to be raised, Xpediter will step into any selected UCH invoked by the percolation request, regardless of the status of the UCH issuing that request.
UCH corrects condition, issues RESUME.	• When a statement is being stepped and causes a condition to be raised, Xpediter will step back into the module being resumed into if it is selected. • If the module being resumed into is not selected, Xpediter will position after the call in the next higher selected module or mainline program. For example, if a mainline program call to an unselected subroutine is stepped, and the subroutine causes an abend, the UCH will get control. If the UCH resumes into that unselected subroutine, Xpediter will position back into the mainline program at the statement just after the call to the subroutine. If the subroutine had been selected, Xpediter would have positioned at the resume location in the subroutine.

Using the CSECT Selections Screen

If you know the name of the load module and CSECT that you want to test, the CSECT Selections screen (2.6.2) (Figure 5-13) lets you directly enter a load module name and CSECT name. The CSECT offset and length are automatically determined by

Xpediter/CICS. This screen also provides a summary of all CSECTs that you have selected from the List of CSECTs screen (2.6.1) as shown in Figure 5-3 on page 5-3.

Figure 5-13. CSECT Selections Screen (2.6.2)



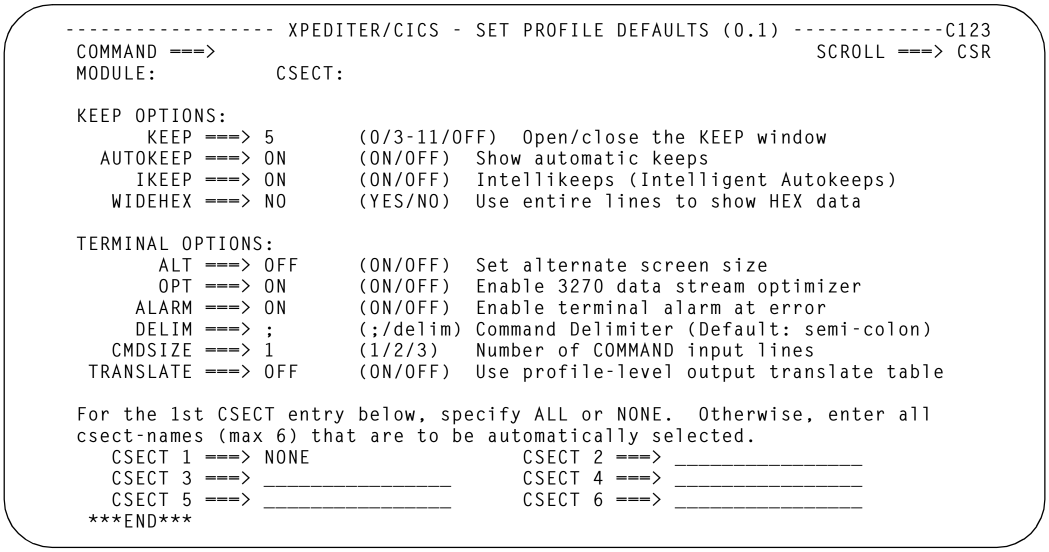
```
------------------ XPEDITER/CICS - CSECT SELECTIONS (2.6.2) -------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE:          CSECT:
                                                                 ENTRY 000001
DEL   MODULE     CSECT                LISTING    OFFSET      LENGTH    ADDRESS
-----------------------------------------------------------------------------
 _    CWDEMPE    CWCDSUBA             CWCDSUBA   00001AB8   000000C8   0B0BDEB8
 _    ________   ________________     ________   ________   ________
 _    ________   ________________     ________   ________   ________
 _    ________   ________________     ________   ________   ________
```

Using Profiles to Select CSECTs

If you know you will be working with some CSECTs through many test sessions, you can set up a profile that will select those CSECTs automatically whenever you enter Xpediter/CICS.

1. Type **XPED 0.1** on a blank CICS screen and press Enter to display the Set Profile Defaults screen (0.1).
2. Press PF8 to display the next page of the Set Profile Defaults screen (0.1) (Figure 5-14).

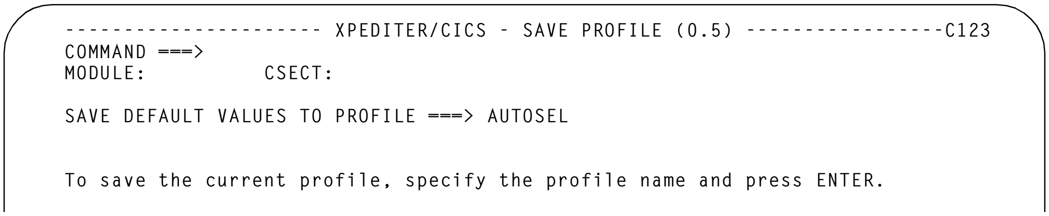
Figure 5-14. Set Profile Defaults Screen (0.1)



```
----------------- XPEDITER/CICS - SET PROFILE DEFAULTS (0.1) ------------C123
 COMMAND ===>                                                  SCROLL ===> CSR
 MODULE:          CSECT:

 KEEP OPTIONS:
       KEEP ===> 5       (0/3-11/OFF)  Open/close the KEEP window
   AUTOKEEP ===> ON      (ON/OFF)  Show automatic keeps
      IKEEP ===> ON      (ON/OFF)  Intellikeeps (Intelligent Autokeeps)
    WIDEHEX ===> NO      (YES/NO)  Use entire lines to show HEX data

 TERMINAL OPTIONS:
        ALT ===> OFF     (ON/OFF)  Set alternate screen size
        OPT ===> ON      (ON/OFF)  Enable 3270 data stream optimizer
      ALARM ===> ON      (ON/OFF)  Enable terminal alarm at error
      DELIM ===> ;       (;/delim) Command Delimiter (Default: semi-colon)
    CMDSIZE ===> 1       (1/2/3)   Number of COMMAND input lines
  TRANSLATE ===> OFF     (ON/OFF)  Use profile-level output translate table

 For the 1st CSECT entry below, specify ALL or NONE.  Otherwise, enter all
 csect-names (max 6) that are to be automatically selected.
    CSECT 1 ===> NONE                 CSECT 2 ===> ________________
    CSECT 3 ===> ________________     CSECT 4 ===> ________________
    CSECT 5 ===> ________________     CSECT 6 ===> ________________
  ***END***
```

3. Type **CWCDSUBA** over the NONE value in the CSECT 1 field.
4. Press Enter to update the field.
5. Save the profile by typing **=0.5** in the COMMAND field and pressing Enter. The Save Profile screen (0.5) is displayed (Figure 5-15).

Figure 5-15. Save Profile Screen (0.5)



```
---------------------- XPEDITER/CICS - SAVE PROFILE (0.5) -----------------C123
COMMAND ===>
MODULE:          CSECT:

SAVE DEFAULT VALUES TO PROFILE ===> AUTOSEL


To save the current profile, specify the profile name and press ENTER.
```

6. Type **AUTOSEL** in the SAVE DEFAULT VALUES TO PROFILE field and press Enter to save the profile.
7. To load the profile for your next debugging session, enter **XPED P=AUTOSEL** on a blank CICS screen. The profile is loaded, and the CWCDSUBA CSECT is automatically selected.

   This technique can also be used to set the CSECT field to ALL, selecting *all* CSECTs when this profile is loaded.

Chapter 6.
Analyzing Program Execution

This chapter shows how to analyze the execution of a program in order to uncover loops and dead code as well as validate logic paths. It discusses how to prepare for analysis of a program, execute the program, then analyze the data.

The COUNT command is used to designate portions of the application code you want to analyze. As each designated statement is executed, a counter is incremented. After your test is completed, Xpediter/CICS examines the counters and provides statistics about the executed code.

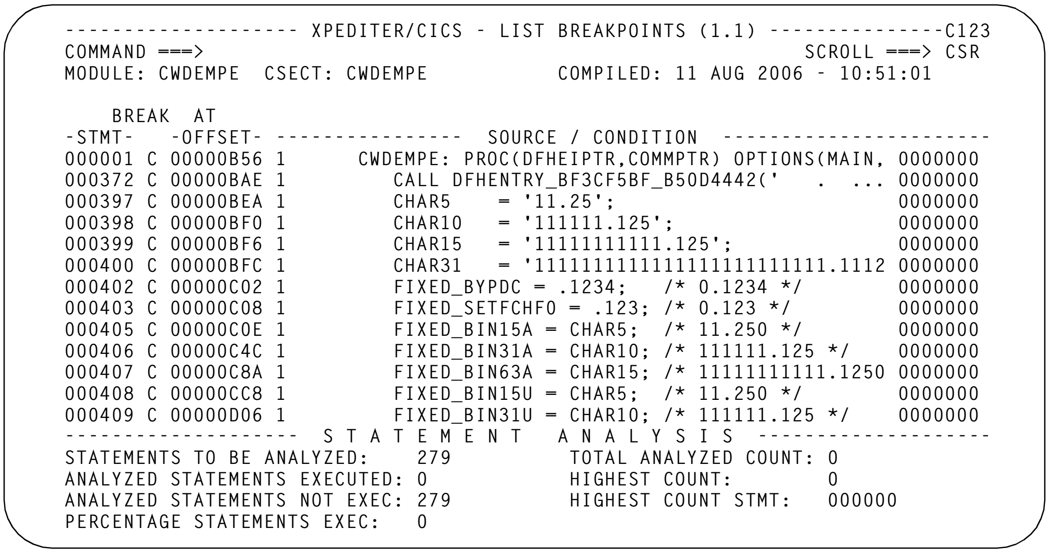
Note: The demonstrations in this chapter should be performed in Xpediter's standard operating mode — ***not*** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

Setting Up the Analysis

1. Type **XPED CWDEMPE** in the upper left corner of a blank CICS screen and press Enter.
2. Type **COUNT ALL** in the COMMAND field and press Enter.

   An informational message will appear verifying that counts have been set.
3. Type **SET FOOT ANALYZE** and press Enter to display the ANALYZE footing. When you are done using FOOT ANALYZE (or any FOOT command), turn it off by typing **SET FOOT OFF** and pressing Enter.
4. Type **=1.1** in the COMMAND field and press Enter to display the List Breakpoints screen (1.1) (Figure 6-1).

Figure 6-1. Setting Up Analysis on the List Breakpoints Screen (1.1)



```
-------------------- XPEDITER/CICS - LIST BREAKPOINTS (1.1) ---------------C123
COMMAND ===>                                                      SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 11 AUG 2006 - 10:51:01

    BREAK  AT
-STMT-   -OFFSET- ----------------  SOURCE / CONDITION  -----------------------
000001 C 00000B56 1      CWDEMPE: PROC(DFHEIPTR,COMMPTR) OPTIONS(MAIN, 0000000
000372 C 00000BAE 1         CALL DFHENTRY_BF3CF5BF_B50D4442('   .  ... 0000000
000397 C 00000BEA 1         CHAR5    = '11.25';                        0000000
000398 C 00000BF0 1         CHAR10   = '111111.125';                   0000000
000399 C 00000BF6 1         CHAR15   = '11111111111.125';              0000000
000400 C 00000BFC 1         CHAR31   = '1111111111111111111111111.1112 0000000
000402 C 00000C02 1         FIXED_BYPDC = .1234;   /* 0.1234 */        0000000
000403 C 00000C08 1         FIXED_SETFCHF0 = .123; /* 0.123 */         0000000
000405 C 00000C0E 1         FIXED_BIN15A = CHAR5;  /* 11.250 */        0000000
000406 C 00000C4C 1         FIXED_BIN31A = CHAR10; /* 111111.125 */    0000000
000407 C 00000C8A 1         FIXED_BIN63A = CHAR15; /* 11111111111.1250 0000000
000408 C 00000CC8 1         FIXED_BIN15U = CHAR5;  /* 11.250 */        0000000
000409 C 00000D06 1         FIXED_BIN31U = CHAR10; /* 111111.125 */    0000000
--------------------  S T A T E M E N T   A N A L Y S I S  -------------------
STATEMENTS TO BE ANALYZED:    279          TOTAL ANALYZED COUNT: 0
ANALYZED STATEMENTS EXECUTED: 0            HIGHEST COUNT:        0
ANALYZED STATEMENTS NOT EXEC: 279          HIGHEST COUNT STMT:   000000
PERCENTAGE STATEMENTS EXEC:   0
```

This screen shows the statements that were set to be analyzed, as well as the statistics concerning the execution. The STATEMENTS TO BE ANALYZED field is set to 279, indicating that there are 279 executable statements in the program in this example. The ANALYZED STATEMENTS NOT EXEC field is set to 279 because the program has not been executed yet. All other values are set to 0 for the same reason. Notice that the COUNT field for each statement is set to 0.

Executing the Program

1. Press Clear to return to CICS and execute the program.
2. Type **XPLE** in the upper left corner of the screen.
3. Press Enter to display the XPLE Demonstration Transaction screen.

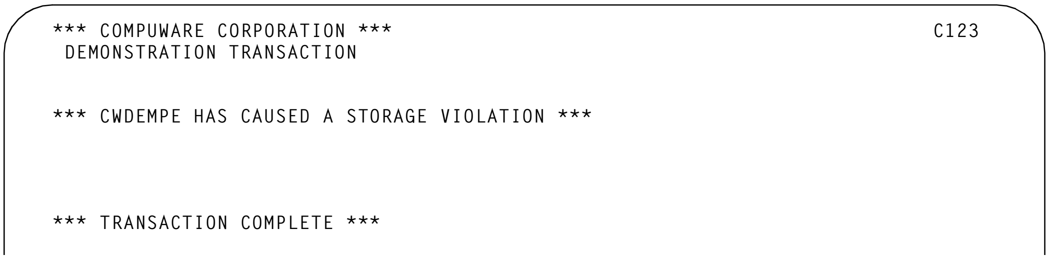
CAUTION:

The next step causes a storage violation. Before performing that step, you should make sure your CICS region is not configured to terminate in response to storage violations.

4. Type **00333** and press Enter. The XPLE message screen appears, indicating that a storage violation has occurred (Figure 6-2).

   In this example, Xpediter/CICS allowed a storage violation to occur because storage protection was turned OFF. The storage violation option was used because it provides a graphic example of a looping problem.

Figure 6-2. Storage Violation on the Demonstration Transaction Screen



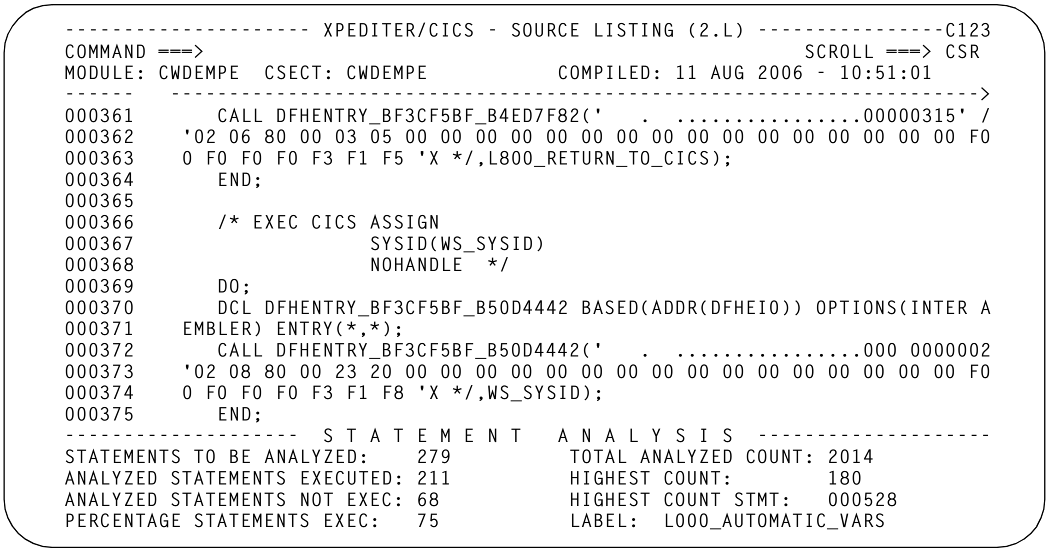
```
*** COMPUWARE CORPORATION ***                                          C123
 DEMONSTRATION TRANSACTION


*** CWDEMPE HAS CAUSED A STORAGE VIOLATION ***




*** TRANSACTION COMPLETE ***
```

Analyzing the Data

1. Press Clear.
2. Type **XPED CWDEMPE** from a cleared CICS screen and press Enter to display the Source Listing screen (2.L).

   The STATEMENT ANALYSIS area displays statistics for the last execution of CWDEMPE, as shown in Figure 6-3.

Figure 6-3. Showing Statistics For Analysis on the Source Listing Screen (2.L)



```
--------------------- XPEDITER/CICS - SOURCE LISTING (2.L) -----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 11 AUG 2006 - 10:51:01
------   ---------------------------------------------------------------------->
000361       CALL DFHENTRY_BF3CF5BF_B4ED7F82('   .  ................00000315' /
000362    '02 06 80 00 03 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0
000363    0 F0 F0 F0 F3 F1 F5 'X */,L800_RETURN_TO_CICS);
000364       END;
000365
000366       /* EXEC CICS ASSIGN
000367                    SYSID(WS_SYSID)
000368                    NOHANDLE  */
000369       DO;
000370       DCL DFHENTRY_BF3CF5BF_B50D4442 BASED(ADDR(DFHEI0)) OPTIONS(INTER A
000371    EMBLER) ENTRY(*,*);
000372       CALL DFHENTRY_BF3CF5BF_B50D4442('   .  ................000 0000002
000373    '02 08 80 00 23 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0
000374    0 F0 F0 F0 F3 F1 F8 'X */,WS_SYSID);
000375       END;
--------------------  S T A T E M E N T   A N A L Y S I S  --------------------
STATEMENTS TO BE ANALYZED:    279          TOTAL ANALYZED COUNT: 2014
ANALYZED STATEMENTS EXECUTED: 211          HIGHEST COUNT:        180
ANALYZED STATEMENTS NOT EXEC: 68           HIGHEST COUNT STMT:   000528
PERCENTAGE STATEMENTS EXEC:   75           LABEL:  L000_AUTOMATIC_VARS
```

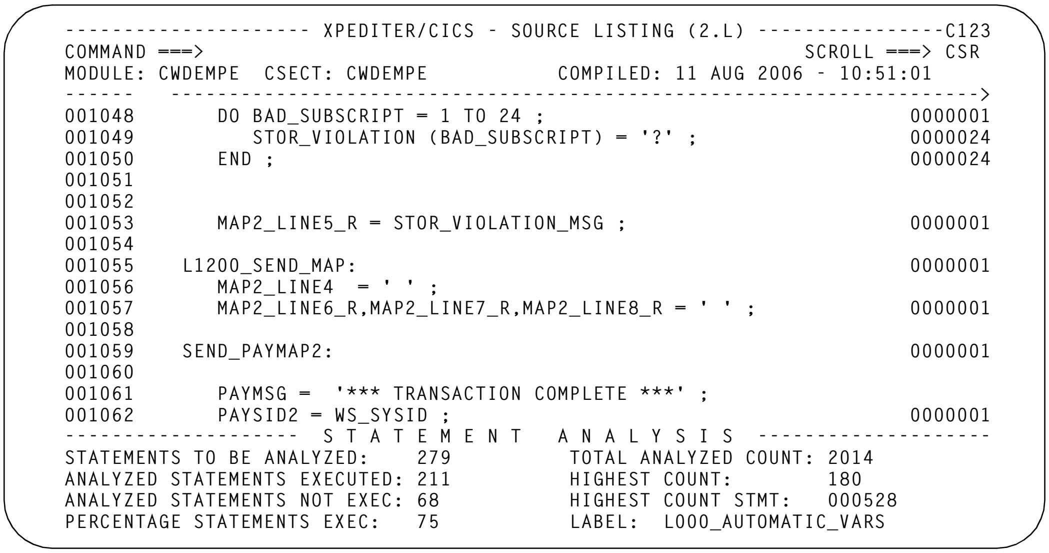
In this example, the statistics in the STATEMENT ANALYSIS area have changed to show the number of statements that were executed.

This area shows that 279 statements were set for analysis. Of these, 211 were executed and 68 were not. The 211 statements were executed a total of 2014 times (TOTAL ANALYZED COUNT), with one executing 180 times (HIGHEST COUNT). This indicates a loop in the program. HIGHEST COUNT STMT points to the statement number where the loop occurred. LABEL shows the name of the paragraph that contains that statement.

Also note that the COUNT parameters have changed. The first statement was executed twice: once to send the initial XPLE screen and once to process the information on that screen.

3. Type **L 1048** in the COMMAND field to locate statement 1048 (statement number just before where the loop occurred) and press Enter to see where and why the storage violation took place (Figure 6-4).

Figure 6-4. Browsing Source Code on the Source Listing Screen (2.L)



```
--------------------- XPEDITER/CICS - SOURCE LISTING (2.L) -----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 11 AUG 2006 - 10:51:01
------   ---------------------------------------------------------------------->
001048       DO BAD_SUBSCRIPT = 1 TO 24 ;                                0000001
001049          STOR_VIOLATION (BAD_SUBSCRIPT) = '?' ;                   0000024
001050       END ;                                                       0000024
001051
001052
001053       MAP2_LINE5_R = STOR_VIOLATION_MSG ;                         0000001
001054
001055    L1200_SEND_MAP:                                                0000001
001056       MAP2_LINE4  = ' ' ;
001057       MAP2_LINE6_R,MAP2_LINE7_R,MAP2_LINE8_R = ' ' ;              0000001
001058
001059    SEND_PAYMAP2:                                                  0000001
001060
001061       PAYMSG =  '*** TRANSACTION COMPLETE ***' ;
001062       PAYSID2 = WS_SYSID ;                                        0000001
--------------------  S T A T E M E N T   A N A L Y S I S  --------------------
STATEMENTS TO BE ANALYZED:    279          TOTAL ANALYZED COUNT: 2014
ANALYZED STATEMENTS EXECUTED: 211          HIGHEST COUNT:        180
ANALYZED STATEMENTS NOT EXEC: 68           HIGHEST COUNT STMT:   000528
PERCENTAGE STATEMENTS EXEC:   75           LABEL:  L000_AUTOMATIC_VARS
```

In this case, the storage violation occurred because the subscript is being checked for a maximum value greater than 24, while the table has only 16 entries.

You can use the FIND COUNT command to browse through the source listing and examine the COUNT associated with each statement. You can also use the SHOW COUNT command to display the lines selected for analysis. By looking at these counts, you can easily see how your program is processing. If Xpediter’s trace function has been activated, you can also view the Program Trace screen (2.4) to review the logic flow.

You can reset the analysis by entering a new COUNT command. It can be turned OFF by entering the DELETE COUNT command.

You can also use the COUNT command to check the overall efficiency of your program. Use the COUNT ALL command to set up an analysis for the entire program, then execute each logic path in the program without resetting the analysis. Review the STATEMENT ANALYSIS to determine if any code has not been executed. By leaving the analysis active and executing all logic paths in the program, you can easily pinpoint code that was not executed.

Remember to end the session as described in “Exiting Xpediter/CICS and Ending a Debugging Session” on page 2-6.

Chapter 7. Monitoring Tasks Started from Remote Terminals

This chapter describes how to use the Trap Summary screen (1.6) to monitor remote terminals and non-terminal related tasks. The Trap Summary screen assigns a master terminal to be used to trap abends occurring at other terminals or in non-terminal tasks.

The first four sections in this chapter assume that an end user is experiencing a problem with a program. A person in the systems group will monitor the program for abends. These sections show how to set remote traps and view a remote session, how to defer remote trap selection, and how to release trapped terminals. The last section explains the technique you should use when debugging programs that are not related to terminals.

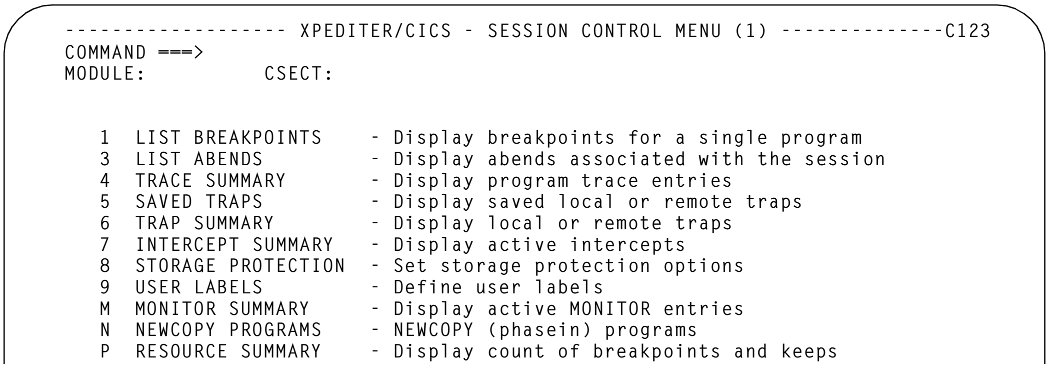
Note: For information on debugging MRO and ISC transactions, distributed transaction processing, and distributed program link, refer to the *Xpediter/CICS Reference Manual*.

Note: The demonstrations in this chapter should be performed in Xpediter/CICS's standard operating mode — ***not*** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

Setting Remote Traps

1. On a blank CICS screen, type **XPED** and press Enter. The Xpediter/CICS Primary Menu will be displayed.
2. Select the session control option by typing **1** in the COMMAND field and pressing Enter. The Session Control Menu appears (Figure 7-1).

Figure 7-1. Session Control Menu



```
------------------ XPEDITER/CICS - SESSION CONTROL MENU (1) --------------C123
COMMAND ===>
MODULE:          CSECT:


   1  LIST BREAKPOINTS    - Display breakpoints for a single program
   3  LIST ABENDS         - Display abends associated with the session
   4  TRACE SUMMARY       - Display program trace entries
   5  SAVED TRAPS         - Display saved local or remote traps
   6  TRAP SUMMARY        - Display local or remote traps
   7  INTERCEPT SUMMARY   - Display active intercepts
   8  STORAGE PROTECTION  - Set storage protection options
   9  USER LABELS         - Define user labels
   M  MONITOR SUMMARY     - Display active MONITOR entries
   N  NEWCOPY PROGRAMS    - NEWCOPY (phasein) programs
   P  RESOURCE SUMMARY    - Display count of breakpoints and keeps
```

3. Type **6** in the COMMAND field and press Enter. The Trap Summary screen (1.6) appears. This screen is used to specify the terminals to be monitored for abends and/or breakpoints.

   The display on your screen will show an entry for your terminal. When the XPED and XPRT transactions are used, an abend trap is automatically set for the terminal on which these transactions are entered.

In this example, you know the transaction that is causing the problem. Set an abend trap for a remote terminal.

4. Type **ALL** in the NETNAME and TERM fields.
5. Type **XPLE** in the TRAN field.
6. Type an asterisk (*) in the PROGRAM field and press Enter to specify that all programs involved in transaction XPLE are monitored.

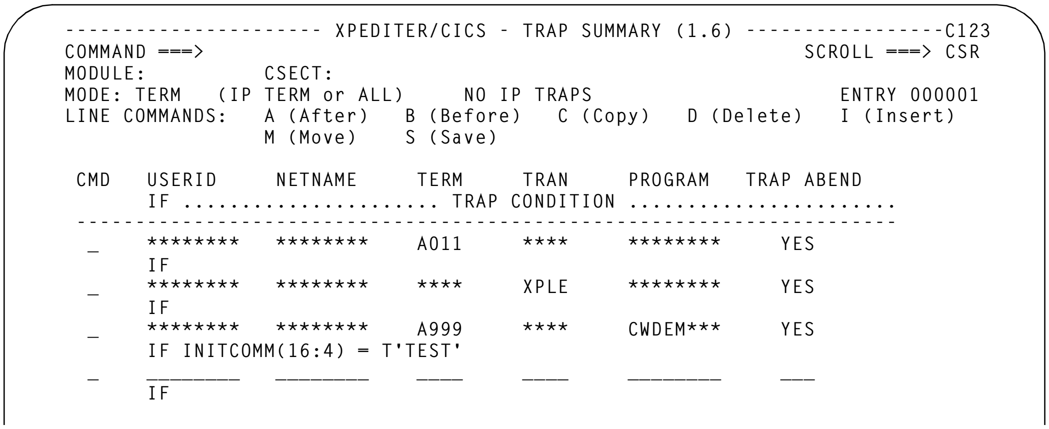
   In Figure 7-2, for example, three traps have been set:

   – For any transaction entered from terminal A011.

   – For XPLE transaction regardless of its origin. This means that all transactions and programs executed at A011 are monitored for abends, as well as any programs involved in the transaction XPLE. If abend situations occur, the information is displayed on your terminal.

   – The third trap is an enhanced trap. If an abend occurs in any program beginning with CWDEM running at terminal A999, and the initial commarea associated with the task contains the string "test" starting in position 16 for a length of 4, the information is displayed on your terminal. Also, any breakpoints in programs beginning with CWDEM will be taken if the initial commarea meets the criteria above and the transaction was started on terminal A999.
7. Press Clear to return to a blank CICS screen.

Figure 7-2. Setting a Trap on the Trap Summary Screen (1.6)



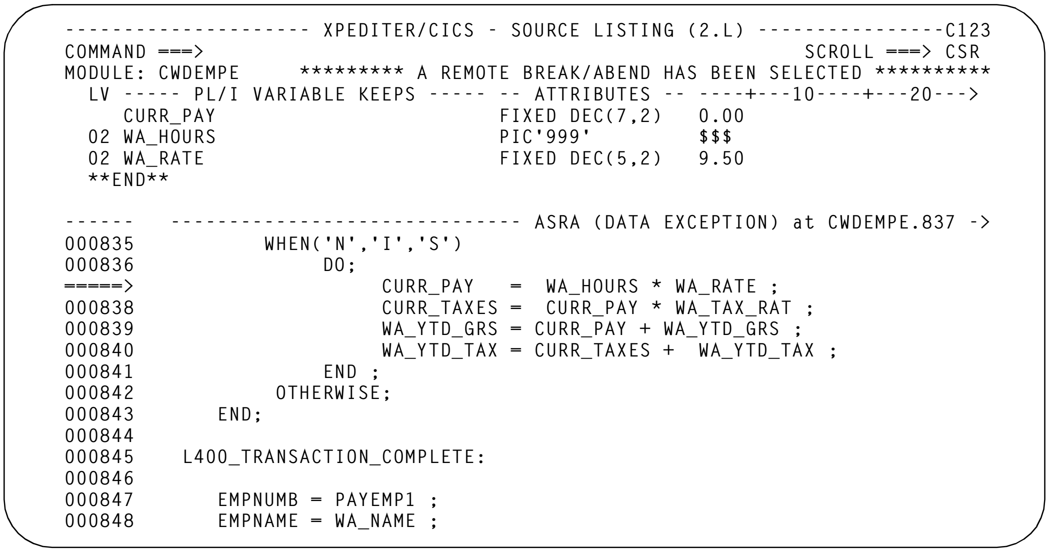
```
---------------------- XPEDITER/CICS - TRAP SUMMARY (1.6) -----------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE:           CSECT:
MODE: TERM   (IP TERM or ALL)     NO IP TRAPS                    ENTRY 000001
LINE COMMANDS:    A (After)   B (Before)   C (Copy)   D (Delete)   I (Insert)
                  M (Move)    S (Save)

 CMD   USERID     NETNAME     TERM     TRAN     PROGRAM   TRAP ABEND
       IF ...................... TRAP CONDITION .......................
 ---------------------------------------------------------------------
  _    ********   ********    A011     ****     ********     YES
       IF
  _    ********   ********    ****     XPLE     ********     YES
       IF
  _    ********   ********    A999     ****     CWDEM***     YES
       IF INITCOMM(16:4) = T'TEST'
  _    ________   ________    ____     ____     ________     ___
       IF
```

Viewing a Remote Session

Start the demonstration transaction from ***another terminal***.

1. Log on to the CICS region at another terminal.
2. Type **XPLE** on a blank CICS screen and press Enter. The Demonstration Transaction screen is displayed.
3. Type **00001** and press Enter to cause an ASRA abend. The terminal is suspended.
4. Return to the original terminal. The Source Listing screen (2.L) (Figure 7-3) is displayed showing CWDEMPE with a message that a remote abend has been selected.

Figure 7-3. Reviewing a Remote Task on the Source Listing Screen

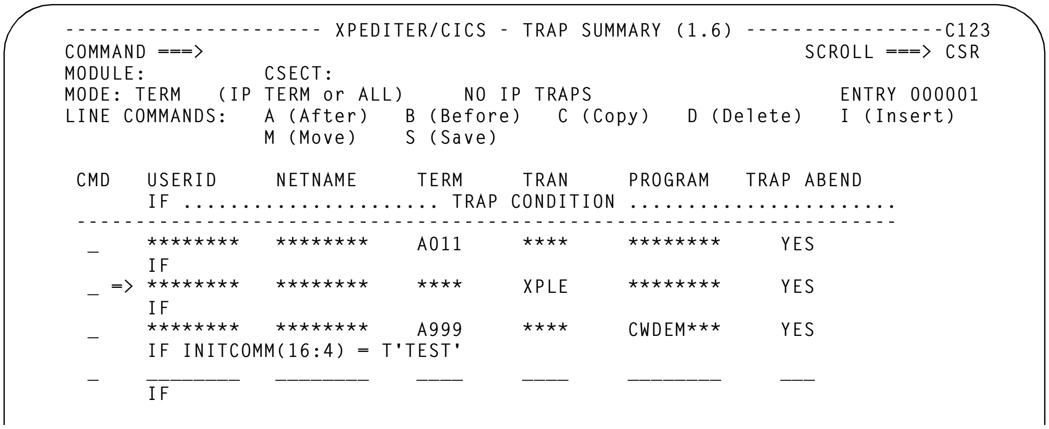


```
--------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                     SCROLL ===> CSR
MODULE: CWDEMPE     ********* A REMOTE BREAK/ABEND HAS BEEN SELECTED **********
  LV ----- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
     CURR_PAY                        FIXED DEC(7,2)   0.00
  02 WA_HOURS                        PIC'999'         $$$
  02 WA_RATE                         FIXED DEC(5,2)   9.50
  **END**

------   ------------------------------ ASRA (DATA EXCEPTION) at CWDEMPE.837 ->
000835           WHEN('N','I','S')
000836                DO;
=====>                     CURR_PAY   =  WA_HOURS * WA_RATE ;
000838                     CURR_TAXES =  CURR_PAY * WA_TAX_RAT ;
000839                     WA_YTD_GRS = CURR_PAY + WA_YTD_GRS ;
000840                     WA_YTD_TAX = CURR_TAXES +  WA_YTD_TAX ;
000841                END ;
000842            OTHERWISE;
000843        END;
000844
000845     L400_TRANSACTION_COMPLETE:
000846
000847        EMPNUMB = PAYEMP1 ;
000848        EMPNAME = WA_NAME ;
```

You now have control over the execution of this program. You can set breakpoints, skips, and keeps, view program storage, step through the program, and resume execution at another point. In this demonstration, you fix the data and continue processing.

5. Position the cursor over the **$$$** in WA_HOURS, type **040**, and press Enter.
6. Type **=1.6** in the COMMAND field and press Enter. The Trap Summary screen (1.6) (Figure 7-4 on page 7-3) is displayed. Notice that the second trap entry has a highlighted arrow following the CMD field. This arrow indicates which trap caused the task to be routed to your session.

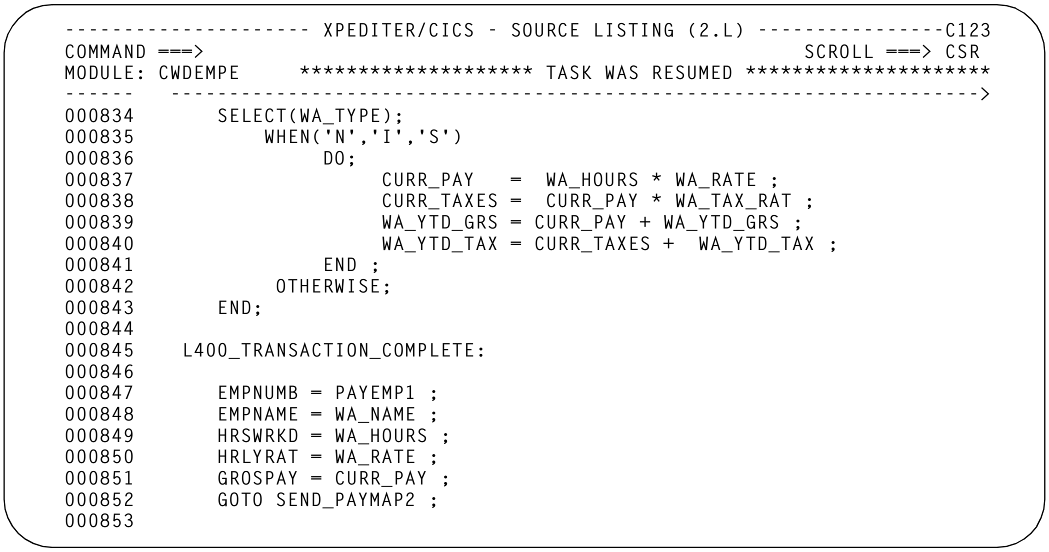
Figure 7-4. Trap Indicated on Trap Summary Screen (1.6)



```
---------------------- XPEDITER/CICS - TRAP SUMMARY (1.6) -----------------C123
COMMAND ===>                                                     SCROLL ===> CSR
MODULE:          CSECT:
MODE: TERM   (IP TERM or ALL)     NO IP TRAPS                       ENTRY 000001
LINE COMMANDS:   A (After)   B (Before)   C (Copy)   D (Delete)   I (Insert)
                 M (Move)    S (Save)

 CMD   USERID     NETNAME     TERM     TRAN     PROGRAM   TRAP ABEND
       IF ..................... TRAP CONDITION ......................
 ---------------------------------------------------------------------
  _    ********   ********    A011     ****     ********     YES
       IF
  _ => ********   ********    ****     XPLE     ********     YES
       IF
  _    ********   ********    A999     ****     CWDEM***     YES
       IF INITCOMM(16:4) = T'TEST'
  _    ________   ________    ____     ____     ________     ___
       IF
```

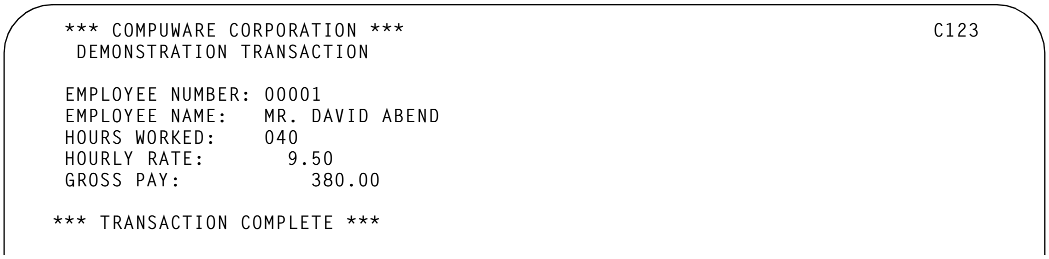
7. Return to the Source Listing screen (2.L) by typing **SOURCE** in the COMMAND field and pressing Enter.
8. Press PF12 (GO) to continue processing. The message **TASK WAS RESUMED** is displayed (Figure 7-5), and control is returned to the user terminal (Figure 7-6).

Figure 7-5. Resuming a Remote Task on the Source Listing Screen (2.L)



```
--------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE: CWDEMPE     ******************** TASK WAS RESUMED ********************
------   ------------------------------------------------------------------->
000834       SELECT(WA_TYPE);
000835           WHEN('N','I','S')
000836                DO;
000837                     CURR_PAY   =  WA_HOURS * WA_RATE ;
000838                     CURR_TAXES =  CURR_PAY * WA_TAX_RAT ;
000839                     WA_YTD_GRS = CURR_PAY + WA_YTD_GRS ;
000840                     WA_YTD_TAX = CURR_TAXES +  WA_YTD_TAX ;
000841                END ;
000842            OTHERWISE;
000843       END;
000844
000845    L400_TRANSACTION_COMPLETE:
000846
000847       EMPNUMB = PAYEMP1 ;
000848       EMPNAME = WA_NAME ;
000849       HRSWRKD = WA_HOURS ;
000850       HRLYRAT = WA_RATE ;
000851       GROSPAY = CURR_PAY ;
000852       GOTO SEND_PAYMAP2 ;
000853
```

Figure 7-6. Demonstration Transaction Screen from the User Terminal



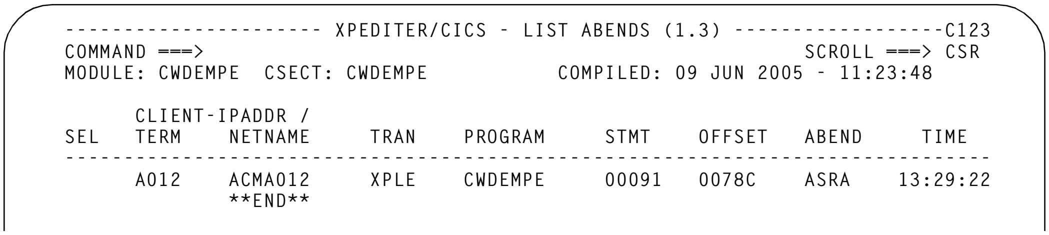
```
 *** COMPUWARE CORPORATION ***                                             C123
  DEMONSTRATION TRANSACTION

 EMPLOYEE NUMBER: 00001
 EMPLOYEE NAME:   MR. DAVID ABEND
 HOURS WORKED:    040
 HOURLY RATE:       9.50
 GROSS PAY:           380.00

*** TRANSACTION COMPLETE ***
```

Deferring Remote Trap Selection

If you set an abend trap for a commonly used transaction, and an abend occurs while you are busy working on a task outside of Xpediter/CICS, you can defer viewing the trap information. When you finish the other task, Xpediter/CICS displays the Source Listing screen (2.L) with a message that an abend has occurred. At this point, you can choose to resolve the abend, or you can defer working on it by pressing Clear and returning to CICS. The abend is still trapped, the user's terminal is suspended, and your terminal is clear to be used for other functions. You can return to the abend at a later time. Note, however, that the user terminal will remain suspended until it is released.

The List Abends screen (1.3) (Figure 7-7) displays the break/abend summary information retained each time a break/abend is trapped. If a remote break/abend is still active, it is highlighted and can be selected by typing an **S** in the SEL column and pressing Enter. The Source Listing screen (2.L) is displayed for that task, and you have control over the execution of the program. Other entries shown on this screen are abends that have been trapped by this terminal during the debugging session.

Figure 7-7. List Abends Screen (1.3)



```
---------------------- XPEDITER/CICS - LIST ABENDS (1.3) ------------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48

      CLIENT-IPADDR /
SEL   TERM    NETNAME     TRAN    PROGRAM     STMT    OFFSET   ABEND     TIME
-------------------------------------------------------------------------------
      A012    ACMA012     XPLE    CWDEMPE     00091   0078C    ASRA    13:29:22
              **END**
```

Releasing Trapped Transactions

Abend traps entered on the Trap Summary screen (1.6) Figure 7-2 on page 7-2 are in effect throughout a debugging session. Xpediter/CICS monitors the user IDs, NETNAMEs, terminals, transactions, and programs for which traps are set until you turn the traps off or end the debugging session. In addition, all abends trapped during your debugging session must be resolved or released before the user's terminal regains control over the program.

You can release the trapped transactions in one of several ways:

- Intercept the trap, fix the problem, and resume the program.
- Temporarily remove your trap and resume the program without fixing the problem.
- Select the trapped task on the List Abends screen (1.3), then access the Exit Session screen (X) and request a dump of the program.
- End the debugging session by entering **=X** from any Xpediter/CICS screen.

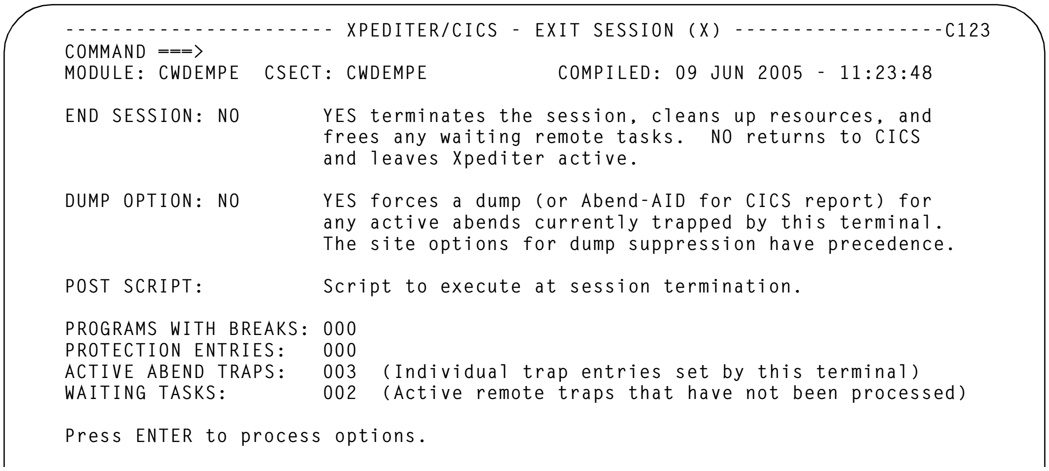
When you end your session while remote abends are waiting and traps are active, all trapped transactions are automatically freed and all abend traps are reset. To end your session:

1. Type **=X** and press Enter. The Exit Session screen (X) (Figure 7-8) appears.

   The lower portion of the Exit Session screen displays a summary of the current testing session. The ACTIVE ABEND TRAPS field lets you know that you could be receiving trap bulletins. The WAITING TASKS field indicates the number of remote abends that have been trapped and have not yet been resumed or terminated. These transactions are suspended until they are freed.

2. Type **Y** in the END SESSION field. If a dump is required, type **Y** in the DUMP OPTION field. If you would like to execute a user or system script, type the script member name in the POST SCRIPT field. Press Enter.

Figure 7-8. Exit Session Screen (X)



```
---------------------- XPEDITER/CICS - EXIT SESSION (X) ------------------C123
COMMAND ===>
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48

END SESSION: NO       YES terminates the session, cleans up resources, and
                      frees any waiting remote tasks.  NO returns to CICS
                      and leaves Xpediter active.

DUMP OPTION: NO       YES forces a dump (or Abend-AID for CICS report) for
                      any active abends currently trapped by this terminal.
                      The site options for dump suppression have precedence.

POST SCRIPT:          Script to execute at session termination.

PROGRAMS WITH BREAKS: 000
PROTECTION ENTRIES:   000
ACTIVE ABEND TRAPS:   003  (Individual trap entries set by this terminal)
WAITING TASKS:        002  (Active remote traps that have not been processed)

Press ENTER to process options.
```

To release a remotely trapped transaction without ending your session, do the following:

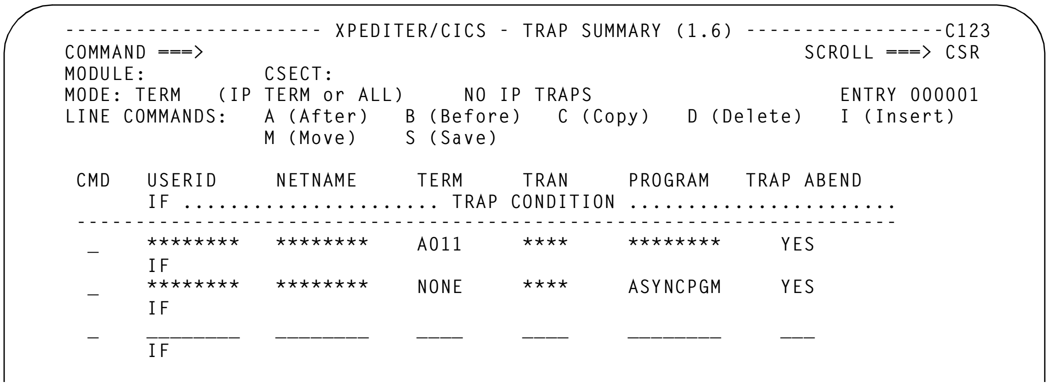
1. Select the transaction to be released on the List Abends screen (1.3).
2. Type **=X** and press Enter. The Exit Session screen (X) (Figure 7-8) appears.
3. Leave **NO** in the END SESSION field and type **Y** in the DUMP OPTION field. Press Enter. With the DUMP OPTION field set to YES, Xpediter/CICS will generate a CICS transaction dump and free the currently selected trapped transaction.

Note: If you clear the screen or press Enter on the Exit Session screen (X) with **NO** in the END SESSION and DUMP OPTION fields, any remotely trapped transactions will not be freed.

Viewing Traps for Asynchronous Transactions

The second entry shown in Figure 7-9 traps abends that occur only in asynchronous transactions executing program ASYNCPGM.

Figure 7-9. Trap for Asynchronous Tasks on the Trap Summary Screen (1.6)



```
--------------------- XPEDITER/CICS - TRAP SUMMARY (1.6) -----------------C123
COMMAND ===>                                                 SCROLL ===> CSR
MODULE:          CSECT:
MODE: TERM   (IP TERM or ALL)     NO IP TRAPS                      ENTRY 000001
LINE COMMANDS:   A (After)   B (Before)   C (Copy)   D (Delete)   I (Insert)
                 M (Move)    S (Save)

 CMD   USERID     NETNAME     TERM     TRAN     PROGRAM   TRAP ABEND
       IF ...................... TRAP CONDITION .......................
 ----------------------------------------------------------------------
  _    ********   ********    A011     ****     ********     YES
       IF
  _    ********   ********    NONE     ****     ASYNCPGM     YES
       IF
  _    ________   ________    ____     ____     ________     ___
       IF
```

To prepare for debugging an asynchronous task, set a trap as shown in Figure 7-9 to provide abend protection for the program. Also set a breakpoint at the beginning of the program to give you control of the program as soon as it starts.

To set a breakpoint at the beginning of a program, type the program name in the MODULE field in the upper left portion of the screen, type **BEFORE** in the COMMAND field, and press Enter.

Chapter 8. Providing Storage Protection

This chapter discusses the storage protection features of Xpediter/CICS, including setting storage protection and allowing storage violations.

The first two sections show how Xpediter/CICS handles a storage violation and how you can allow processing to continue after a violation has been intercepted. This exercise is done using the XPED transaction.

For more information regarding storage violation protection, refer to the *Xpediter/CICS Reference Manual*.

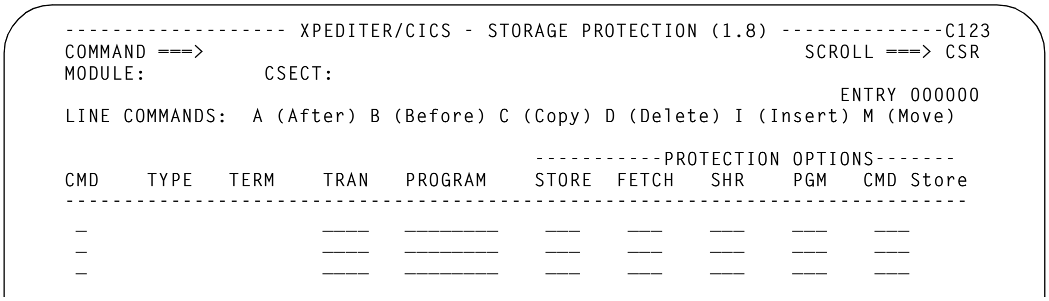
Note: The demonstrations in this chapter should be performed in Xpediter's standard operating mode — ***not*** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

Setting Storage Protection

If you have a program that is ready to go into production, you should test it to make sure it is not causing storage violations. To do this, you use XPED to set up storage protection, and then test the transaction.

1. Type **XPED 1.8** in the upper left corner of a blank CICS screen.
2. Press Enter to display the Storage Protection screen (1.8) (Figure 8-1).

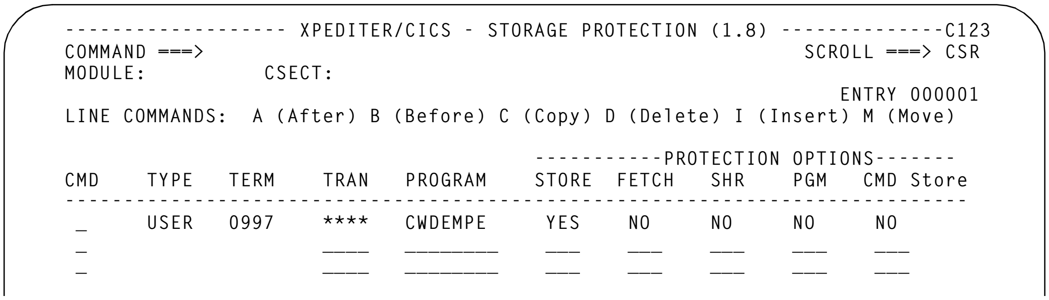
Figure 8-1. Storage Protection Screen (1.8)



```
------------------ XPEDITER/CICS - STORAGE PROTECTION (1.8) -------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
                                                                  ENTRY 000000
LINE COMMANDS:  A (After) B (Before) C (Copy) D (Delete) I (Insert) M (Move)

                                           -----------PROTECTION OPTIONS-------
CMD    TYPE   TERM    TRAN   PROGRAM     STORE  FETCH   SHR    PGM   CMD Store
-----------------------------------------------------------------------------
 _                    ____   ________     ___    ___    ___    ___    ___
 _                    ____   ________     ___    ___    ___    ___    ___
 _                    ____   ________     ___    ___    ___    ___    ___
```

3. Type **CWDEMPE** in the PROGRAM field in the middle of the screen and type **Y** in the STORE field under PROTECTION OPTIONS.
4. Press Enter to enter these values and redisplay the Storage Protection screen (1.8). See Figure 8-2.

Figure 8-2. Storage Protection Screen (1.8) with a Protection Entry



```
------------------ XPEDITER/CICS - STORAGE PROTECTION (1.8) -------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE:         CSECT:
                                                                 ENTRY 000001
LINE COMMANDS:  A (After) B (Before) C (Copy) D (Delete) I (Insert) M (Move)

                                        -----------PROTECTION OPTIONS-------
CMD    TYPE   TERM    TRAN   PROGRAM    STORE  FETCH   SHR    PGM   CMD Store
-----------------------------------------------------------------------------
 _     USER   0997    ****   CWDEMPE     YES    NO     NO     NO     NO
 _                    ____   ________    ___    ___    ___    ___    ___
 _                    ____   ________    ___    ___    ___    ___    ___
```

On the Storage Protection screen (1.8), Xpediter/CICS assigns a value of USER in the TYPE field, indicating that this entry was set up by the user. The current terminal is automatically entered in the TERM field.

The asterisks in the TRAN field indicate that this entry is valid for any transaction that executes the program CWDEMPE. The protection options FETCH, SHR, PGM, and CMD Store are automatically set to NO. These entries are valid during the current debugging session. For more information about these fields, press PF1.

Allowing Storage Violations

1. Press Clear to return to CICS to test the transaction.
2. Type **XPLE** on a blank screen.
3. Press Enter to display the XPLE Demonstration Transaction screen.
4. Type **00333** to cause a storage violation.
5. Press Enter to display the Source Listing screen (2.L) (Figure 8-3). On this screen, Xpediter/CICS displays the messages

   ```
   ****************SUBSCRIPT OUT OF BOUNDS****************
   ```

   and

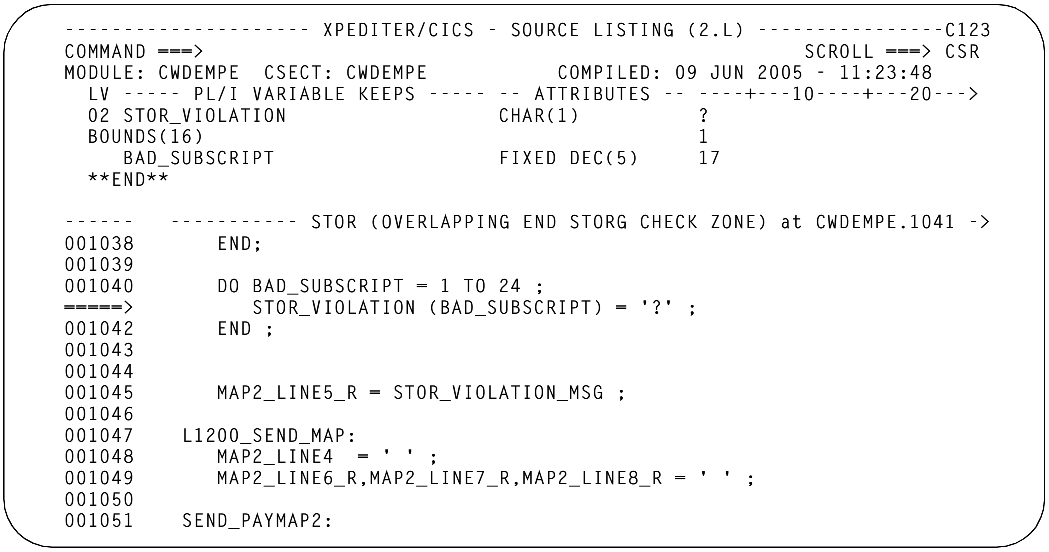
   ```
   ----------- STOR (OVERLAPPING END STORG CHECK ZONE) at CWDEMPE.1041 ->
   ```

   to show that a potential storage violation has been intercepted and prevented.

   When storage protection is turned ON, Xpediter/CICS intercepts any program that attempts to write in a CICS storage area that the program does not own. Xpediter/CICS intercepts all programs that violate CICS storage, but in certain instances, you may not agree with its analysis. The ALLOW command is used for these cases.

   In this example, the user might decide that the subscript is acceptable and that the ALLOW command should be used to permit the storage violation to occur.

Figure 8-3. Source Listing Screen (2.L) - Intercepting a Storage Violation



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ---------------C123
COMMAND ===>                                                     SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
  LV ----- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
  02 STOR_VIOLATION                  CHAR(1)          ?
  BOUNDS(16)                                          1
     BAD_SUBSCRIPT                   FIXED DEC(5)     17
  **END**

------   ----------- STOR (OVERLAPPING END STORG CHECK ZONE) at CWDEMPE.1041 ->
001038       END;
001039
001040       DO BAD_SUBSCRIPT = 1 TO 24 ;
=====>          STOR_VIOLATION (BAD_SUBSCRIPT) = '?' ;
001042       END ;
001043
001044
001045       MAP2_LINE5_R = STOR_VIOLATION_MSG ;
001046
001047    L1200_SEND_MAP:
001048       MAP2_LINE4  = ' ' ;
001049       MAP2_LINE6_R,MAP2_LINE7_R,MAP2_LINE8_R = ' ' ;
001050
001051    SEND_PAYMAP2:
```

Note: Xpediter/CICS is shipped with the ALLOWCM global parameter set to OFF for the XPED transaction. Unless this value has been changed to ON at your site, you will be unable to use the ALLOW command to allow the storage violation described here. Read the remainder of the exercise to understand the concepts presented.

6. Type **ALLOW** in the COMMAND field.
7. Press Enter to redisplay the Source Listing screen (2.L). Depending on the situation, one of three things will happen:
   a. The message

      ```
      ****** THE CURRENT STORAGE VIOL. WILL BE ALLOWED *******
      ```

      is displayed to show that Xpediter/CICS will allow the storage violation to occur. You would continue with the next step.

   b. If the message

      ```
      ****** ALLOW COMMAND IS DISABLED IN GLOBAL TABLE ******
      ```

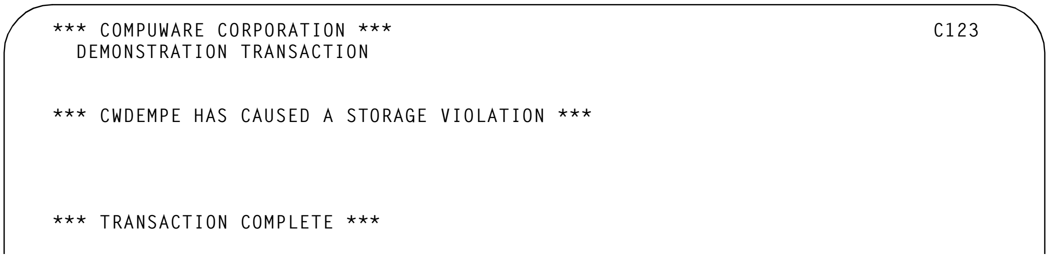
      is displayed, ALLOWCM is set to OFF in the Xpediter/CICS global table, and storage violations will not be allowed. If you want to use the ALLOW command, talk to your site installer to have the ALLOWCM parameter value changed. Read the remainder of the exercise to understand the concepts presented.

   c. If the message

      ```
      ****** NOT ALLOWED TO VIOLATE CICS CONTROL INFO ******
      ```

      is displayed, Xpediter/CICS has prevented you from accidentally overwriting CICS storage check zones and causing a CICS storage violation, even though the ALLOWCM parameter is set to ON. If you examine the program, you will notice that the subscript is, in fact, too large for the area defined. The only way to continue with this example is to manually change the value of BAD_SUBSCRIPT to 24, for example, then use the GOTO command to resume from the next END statement. You can chose to do this before continuing with the next step, or simply read the remainder of the exercise to understand the concepts presented.
8. Press PF12 to resume processing of the program. The program screen shown in Figure 8-4 appears, indicating that a storage violation occurred.

Figure 8-4. Demonstration Transaction Screen: Displaying a Storage Violation



```
*** COMPUWARE CORPORATION ***                                                  C123
  DEMONSTRATION TRANSACTION


*** CWDEMPE HAS CAUSED A STORAGE VIOLATION ***




*** TRANSACTION COMPLETE ***
```

Remember to end the session as described in "Exiting Xpediter/CICS and Ending a Debugging Session" on page 2-6.

Chapter 9. Interfacing with Abend-AID for CICS

This chapter discusses the interface to Abend-AID for CICS, which allows you to access Abend-AID for CICS without leaving Xpediter/CICS.

This exercise assumes that you have completed the examples in Chapter 3, “Testing a PL/I Program”

In this chapter, Xpediter/CICS is turned on to monitor the XPLE transaction, which abends with an AEIM. Abend-AID for CICS is used to help solve the abend.

Note: If your site is using a release of Abend-AID for CICS other than that shown in this chapter, your screens may appear different.

Note: The demonstrations in this chapter should be performed in Xpediter’s standard operating mode — ***not*** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

Turning on Abend-AID for CICS

Abend-AID for CICS must be installed and turned on before interfacing with Xpediter/CICS. If Abend-AID for CICS is already turned on, go to step 1 in the next section. If Abend-AID for CICS is not turned on, complete the following steps:

1. Sign on to a valid CICS region.
2. Type **AAON ON** in the upper left corner of a blank CICS screen.
3. Press Enter. Abend-AID for CICS will display messages similar to the following:

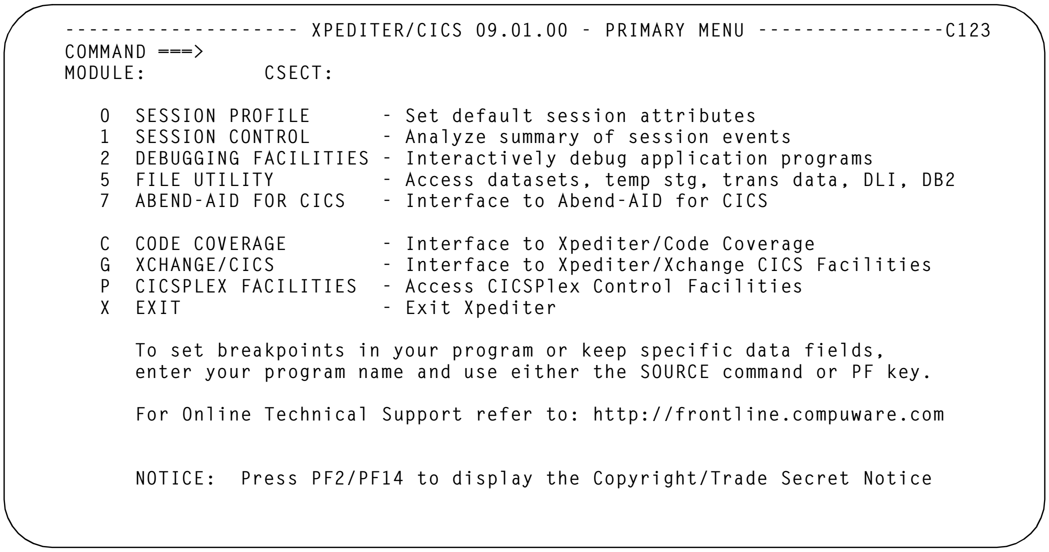
```
CCACI0035I Abend-AID for CICS turned on in region APPLID ACMEC123 at...
CCACI0046I CICS APPLID ACMEC123 SYSID ACM1 connected to view server...
CCACI0074I CICS APPLID ACMEC123 on system ACM1 connected to TDCAS CF45...
```

Refer to the *Abend-AID for CICS Reference Manual* for more information on the use of Abend-AID for CICS.

Accessing Abend-AID for CICS through Xpediter/CICS

1. To access Xpediter/CICS, type **XPED** in the upper left corner of a blank CICS screen.
2. Press Enter to display the Primary Menu shown in Figure 9-1.

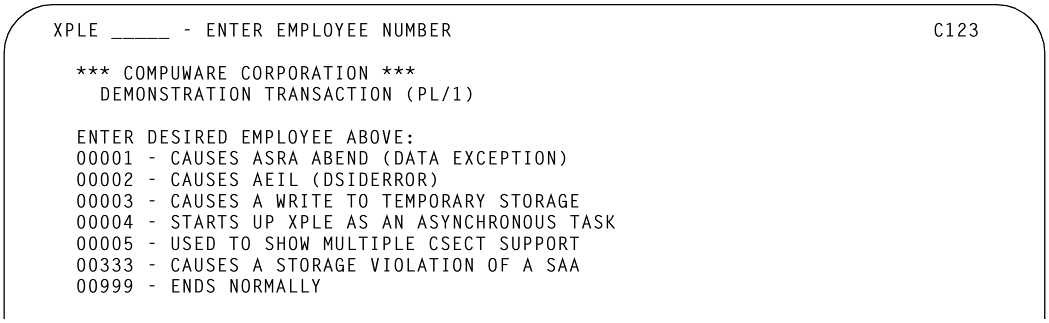
Figure 9-1. Primary Menu (XPED/XPRT)



```
------------------- XPEDITER/CICS 09.01.00 - PRIMARY MENU ---------------C123
COMMAND ===>
MODULE:          CSECT:

   0  SESSION PROFILE      - Set default session attributes
   1  SESSION CONTROL      - Analyze summary of session events
   2  DEBUGGING FACILITIES - Interactively debug application programs
   5  FILE UTILITY         - Access datasets, temp stg, trans data, DLI, DB2
   7  ABEND-AID FOR CICS   - Interface to Abend-AID for CICS

   C  CODE COVERAGE        - Interface to Xpediter/Code Coverage
   G  XCHANGE/CICS         - Interface to Xpediter/Xchange CICS Facilities
   P  CICSPLEX FACILITIES  - Access CICSPlex Control Facilities
   X  EXIT                 - Exit Xpediter

      To set breakpoints in your program or keep specific data fields,
      enter your program name and use either the SOURCE command or PF key.

      For Online Technical Support refer to: http://frontline.compuware.com


      NOTICE:  Press PF2/PF14 to display the Copyright/Trade Secret Notice
```

3. Press Clear. Xpediter/CICS is now set to trap abends.
4. Type **XPLE** and press Enter. The Demonstration Transaction screen is displayed (Figure 9-2).

Figure 9-2. Demonstration Transaction Screen

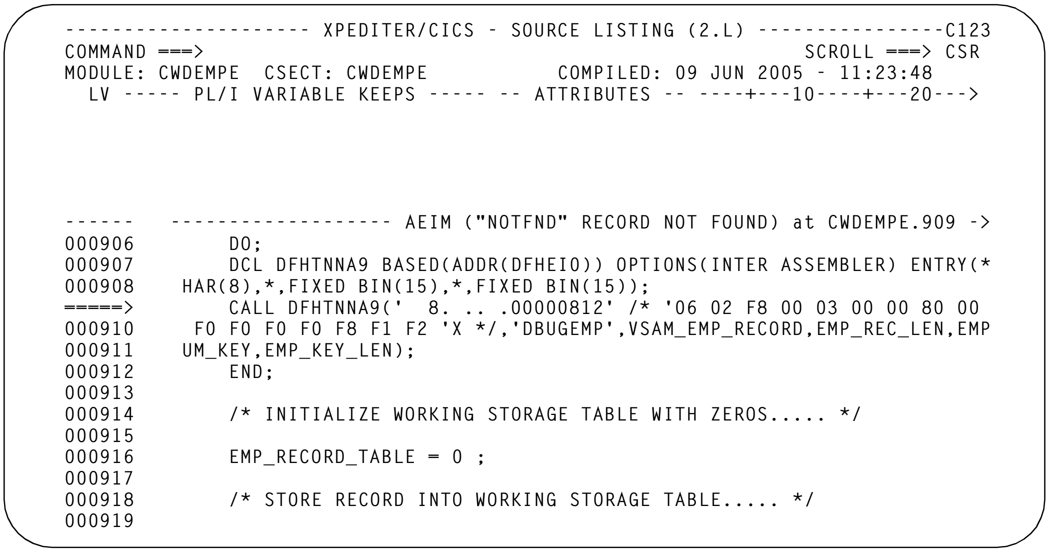


```
XPLE _____ - ENTER EMPLOYEE NUMBER                                         C123

  *** COMPUWARE CORPORATION ***
    DEMONSTRATION TRANSACTION (PL/1)

  ENTER DESIRED EMPLOYEE ABOVE:
  00001 - CAUSES ASRA ABEND (DATA EXCEPTION)
  00002 - CAUSES AEIL (DSIDERROR)
  00003 - CAUSES A WRITE TO TEMPORARY STORAGE
  00004 - STARTS UP XPLE AS AN ASYNCHRONOUS TASK
  00005 - USED TO SHOW MULTIPLE CSECT SUPPORT
  00333 - CAUSES A STORAGE VIOLATION OF A SAA
  00999 - ENDS NORMALLY
```

5. To cause an AEIM abend, type **00002** in the field preceding ENTER EMPLOYEE NUMBER and press Enter. The Source Listing screen (2.L) (Figure 9-3) will be displayed.

   Xpediter/CICS intercepts the abend and reports on the status of the problem. However, you may need more information to resolve the problem. To obtain it, we will access Abend-AID for CICS and issue a snap dump to produce a diagnostic report for this AEIM abend.

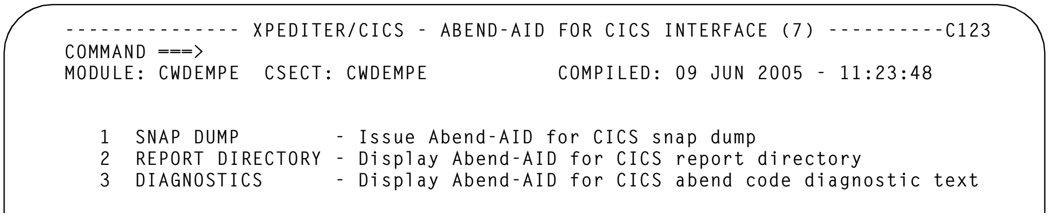
Figure 9-3. Source Listing Screen (2.L)



```
--------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48
  LV ----- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->





------   ------------------- AEIM ("NOTFND" RECORD NOT FOUND) at CWDEMPE.909 ->
000906        DO;
000907        DCL DFHTNNA9 BASED(ADDR(DFHEI0)) OPTIONS(INTER ASSEMBLER) ENTRY(*
000908    HAR(8),*,FIXED BIN(15),*,FIXED BIN(15));
=====>        CALL DFHTNNA9('  8. .. .00000812' /* '06 02 F8 00 03 00 00 80 00
000910     F0 F0 F0 F8 F1 F2 'X */,'DBUGEMP',VSAM_EMP_RECORD,EMP_REC_LEN,EMP
000911    UM_KEY,EMP_KEY_LEN);
000912        END;
000913
000914        /* INITIALIZE WORKING STORAGE TABLE WITH ZEROS..... */
000915
000916        EMP_RECORD_TABLE = 0 ;
000917
000918        /* STORE RECORD INTO WORKING STORAGE TABLE..... */
000919
```

6. Type **MENU** in the COMMAND field and press Enter to redisplay the Xpediter/CICS Primary Menu.
7. Type **7** in the COMMAND field and press Enter to display the Abend-AID for CICS Interface menu (7) (Figure 9-4).

Figure 9-4. Abend-AID for CICS Interface Menu (7)



```
-------------- XPEDITER/CICS - ABEND-AID FOR CICS INTERFACE (7) ----------C123
COMMAND ===>
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48


   1  SNAP DUMP        - Issue Abend-AID for CICS snap dump
   2  REPORT DIRECTORY - Display Abend-AID for CICS report directory
   3  DIAGNOSTICS      - Display Abend-AID for CICS abend code diagnostic text
```

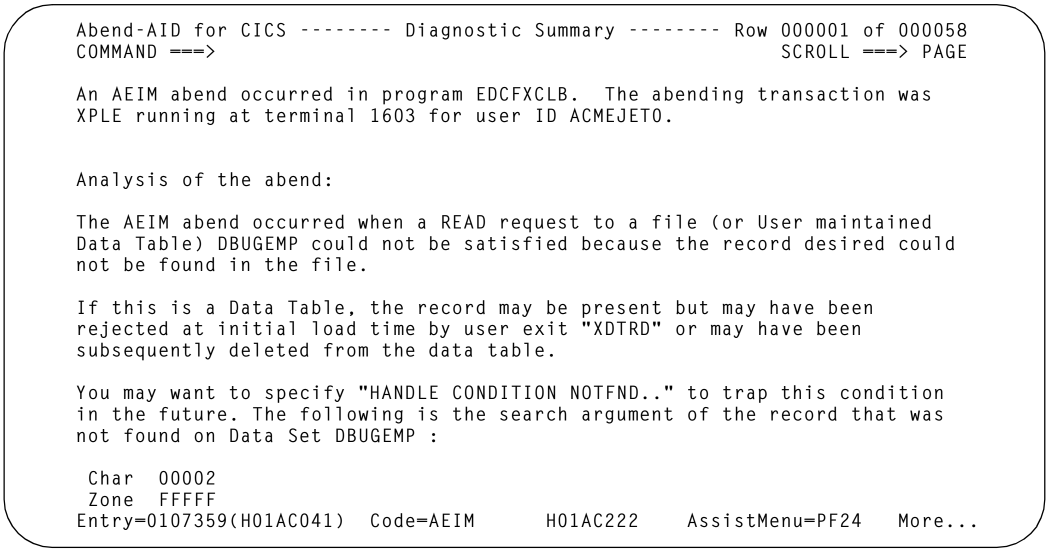
Issuing an Abend-AID for CICS Snap Dump

Abend-AID for CICS is an abend analysis tool that intercepts calls to the dump control program and produces a diagnostic report. The report explains where an abend occurred, why it happened, and how it can be fixed. To get this information, you must issue a snap dump request.

1. To issue a snap dump, type **1** in the COMMAND field of the Abend-AID for CICS Interface Menu (7).
2. Press Enter. A snap dump is taken for the AEIM abend, and Abend-AID for CICS is automatically invoked to create the report for this abend. The Diagnostic Summary of the Abend-AID for CICS report appears as shown in Figure 9-5.

   The Diagnostic Summary gives detailed diagnostics about the trapped abend. The paragraph at the top of the screen identifies the abend code (AEIM), program name (CWDEMPE), abending transaction (XPLE), terminal, and user ID. Additional paragraphs provide a detailed analysis of the abend.

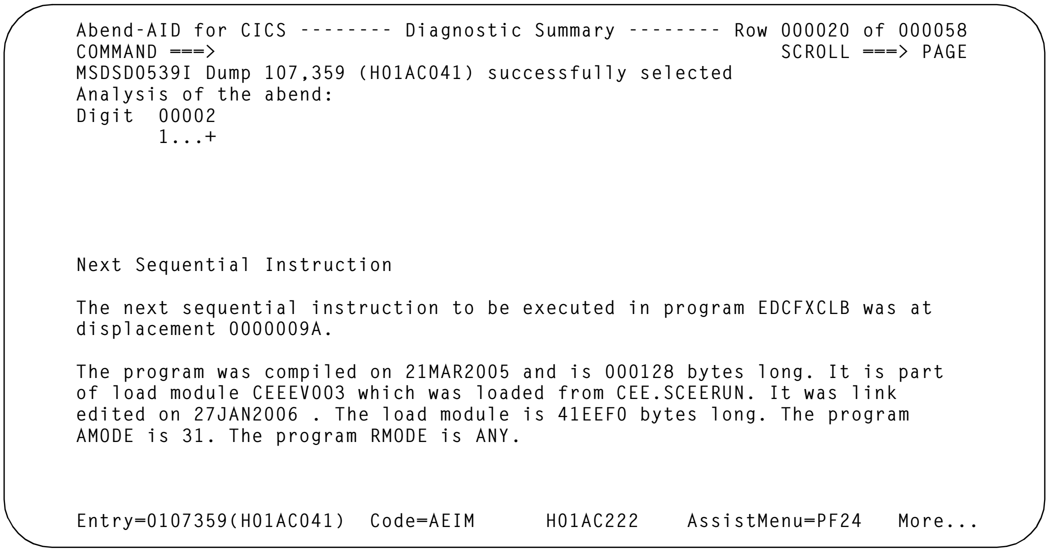
Figure 9-5. Abend-AID for CICS Diagnostic Summary Screen



```
Abend-AID for CICS -------- Diagnostic Summary -------- Row 000001 of 000058
COMMAND ===>                                                SCROLL ===> PAGE

An AEIM abend occurred in program EDCFXCLB.  The abending transaction was
XPLE running at terminal 1603 for user ID ACMEJETO.


Analysis of the abend:

The AEIM abend occurred when a READ request to a file (or User maintained
Data Table) DBUGEMP could not be satisfied because the record desired could
not be found in the file.

If this is a Data Table, the record may be present but may have been
rejected at initial load time by user exit "XDTRD" or may have been
subsequently deleted from the data table.

You may want to specify "HANDLE CONDITION NOTFND.." to trap this condition
in the future. The following is the search argument of the record that was
not found on Data Set DBUGEMP :

 Char  00002
 Zone  FFFFF
Entry=0107359(H01AC041)  Code=AEIM      H01AC222    AssistMenu=PF24   More...
```

3. Press PF8 to scroll through the report. A report example is shown in Figure 9-6.

Figure 9-6. Diagnostic Summary Screen (continued)

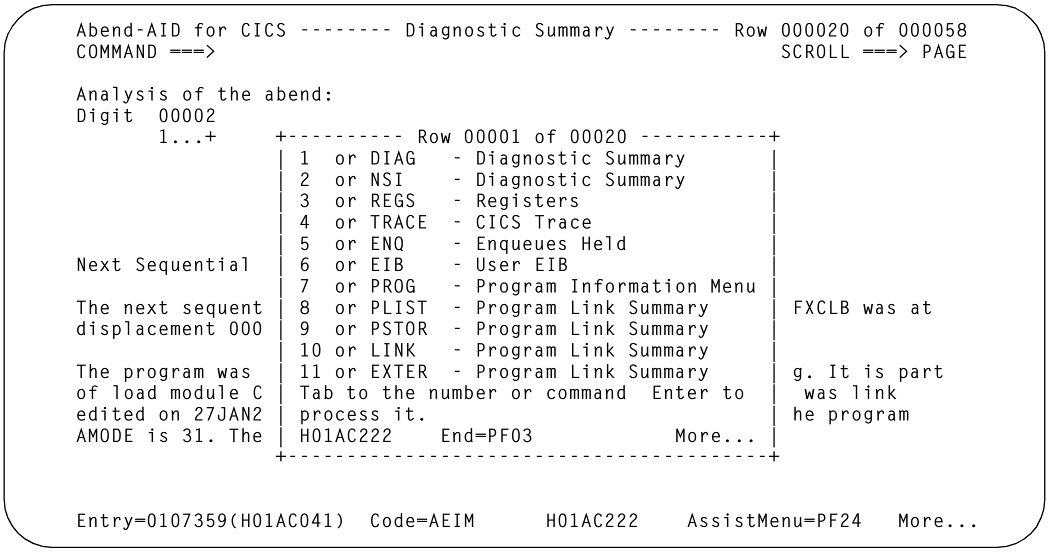


```
Abend-AID for CICS -------- Diagnostic Summary -------- Row 000020 of 000058
COMMAND ===>                                                SCROLL ===> PAGE
MSDSD0539I Dump 107,359 (H01AC041) successfully selected
Analysis of the abend:
Digit  00002
       1...+




Next Sequential Instruction

The next sequential instruction to be executed in program EDCFXCLB was at
displacement 0000009A.

The program was compiled on 21MAR2005 and is 000128 bytes long. It is part
of load module CEEEV003 which was loaded from CEE.SCEERUN. It was link
edited on 27JAN2006 . The load module is 41EEF0 bytes long. The program
AMODE is 31. The program RMODE is ANY.



Entry=0107359(H01AC041)  Code=AEIM      H01AC222    AssistMenu=PF24   More...
```

The Abend-AID for CICS report contains additional information that can be directly accessed by entering the section's number or name in the COMMAND field.

4. To display a menu of the report sections, press PF6. A report menu appears as shown in Figure 9-7.

Figure 9-7. Abend-AID for CICS Report Menu



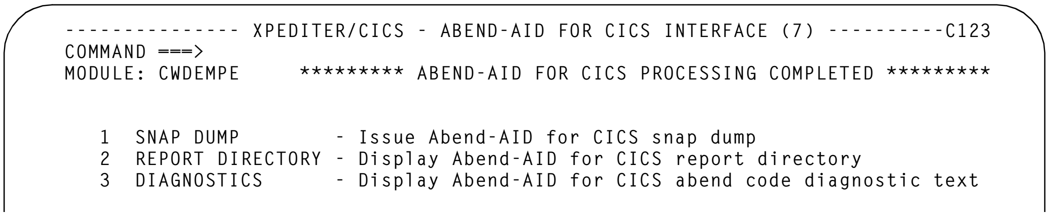
```
Abend-AID for CICS -------- Diagnostic Summary -------- Row 000020 of 000058
COMMAND ===>                                                SCROLL ===> PAGE

Analysis of the abend:
Digit  00002
       1...+     +---------- Row 00001 of 00020 -----------+
                 | 1  or DIAG   - Diagnostic Summary        |
                 | 2  or NSI    - Diagnostic Summary        |
                 | 3  or REGS   - Registers                 |
                 | 4  or TRACE  - CICS Trace                |
                 | 5  or ENQ    - Enqueues Held             |
Next Sequential  | 6  or EIB    - User EIB                  |
                 | 7  or PROG   - Program Information Menu |
The next sequent | 8  or PLIST  - Program Link Summary      | FXCLB was at
displacement 000 | 9  or PSTOR  - Program Link Summary      |
                 | 10 or LINK   - Program Link Summary      |
The program was  | 11 or EXTER  - Program Link Summary      | g. It is part
of load module C | Tab to the number or command  Enter to  |  was link
edited on 27JAN2 | process it.                              | he program
AMODE is 31. The | H01AC222    End=PF03           More... |
                 +------------------------------------------+


Entry=0107359(H01AC041)  Code=AEIM      H01AC222    AssistMenu=PF24   More...
```

Viewing the Abend-AID for CICS Report Directory

The Abend-AID for CICS interface allows an Xpediter/CICS user to access any Abend-AID for CICS report. The Abend-AID for CICS Directory screen contains a list of available reports.

1. To return to the Abend-AID for CICS Interface Menu (7), press PF4 until you are returned to Xpediter/CICS. The menu appears as shown in Figure 9-8.

Figure 9-8. Abend-AID for CICS Interface Menu (7)



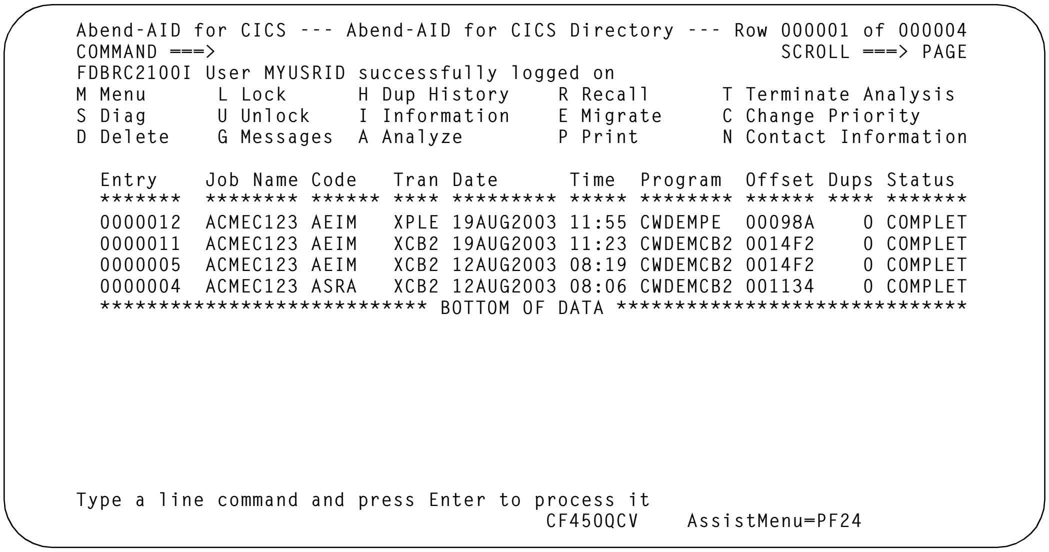
```
--------------- XPEDITER/CICS - ABEND-AID FOR CICS INTERFACE (7) ----------C123
COMMAND ===>
MODULE: CWDEMPE     ********* ABEND-AID FOR CICS PROCESSING COMPLETED *********


   1  SNAP DUMP        - Issue Abend-AID for CICS snap dump
   2  REPORT DIRECTORY - Display Abend-AID for CICS report directory
   3  DIAGNOSTICS      - Display Abend-AID for CICS abend code diagnostic text
```

2. To select the directory, type **2** in the COMMAND field.
3. Press Enter to display the Abend-AID for CICS Directory screen (Figure 9-9).

   This screen displays all abend reports generated for this CICS region. Reports can be selected by entering an **S** to the left of the Entry column next to the desired report.

Note: Your Abend-AID for CICS Directory screen display will differ from the one shown here because abend activity varies from one CICS region to another.

Figure 9-9. Abend-AID for CICS Directory Screen



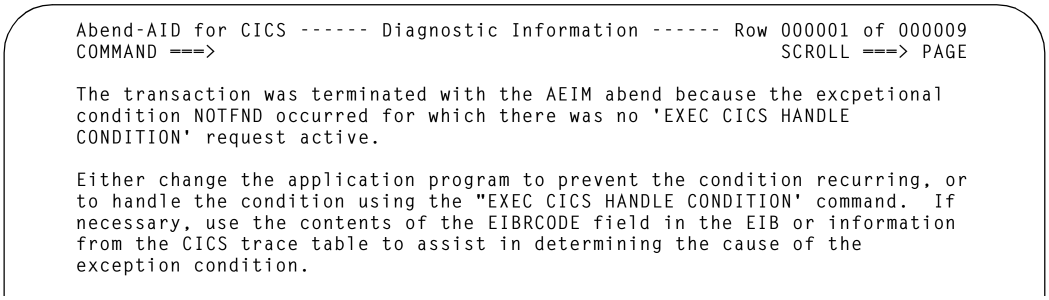
```
Abend-AID for CICS --- Abend-AID for CICS Directory --- Row 000001 of 000004
COMMAND ===>                                              SCROLL ===> PAGE
FDBRC2100I User MYUSRID successfully logged on
M Menu      L Lock      H Dup History    R Recall      T Terminate Analysis
S Diag      U Unlock    I Information    E Migrate     C Change Priority
D Delete    G Messages  A Analyze        P Print       N Contact Information

  Entry    Job Name Code   Tran Date      Time  Program  Offset Dups Status
  *******  ******** ****** **** ********* ***** ******** ****** **** *******
  0000012  ACMEC123 AEIM   XPLE 19AUG2003 11:55 CWDEMPE  00098A    0 COMPLET
  0000011  ACMEC123 AEIM   XCB2 19AUG2003 11:23 CWDEMCB2 0014F2    0 COMPLET
  0000005  ACMEC123 AEIM   XCB2 12AUG2003 08:19 CWDEMCB2 0014F2    0 COMPLET
  0000004  ACMEC123 ASRA   XCB2 12AUG2003 08:06 CWDEMCB2 001134    0 COMPLET
  *************************** BOTTOM OF DATA *****************************







Type a line command and press Enter to process it
                                        CF450QCV    AssistMenu=PF24
```

Viewing the Abend-AID for CICS Diagnostics

The Abend-AID for CICS interface provides additional diagnostics to help solve abends.

1. To return to the Abend-AID for CICS Interface Menu (7), press PF3. The menu appears, and you are now back in Xpediter/CICS.
2. To view the diagnostic messages, type **3** in the COMMAND field.
3. Press Enter to display the Abend-AID for CICS Diagnostic Information screen (Figure 9-10).

Figure 9-10. Abend-AID for CICS Diagnostic Information Screen

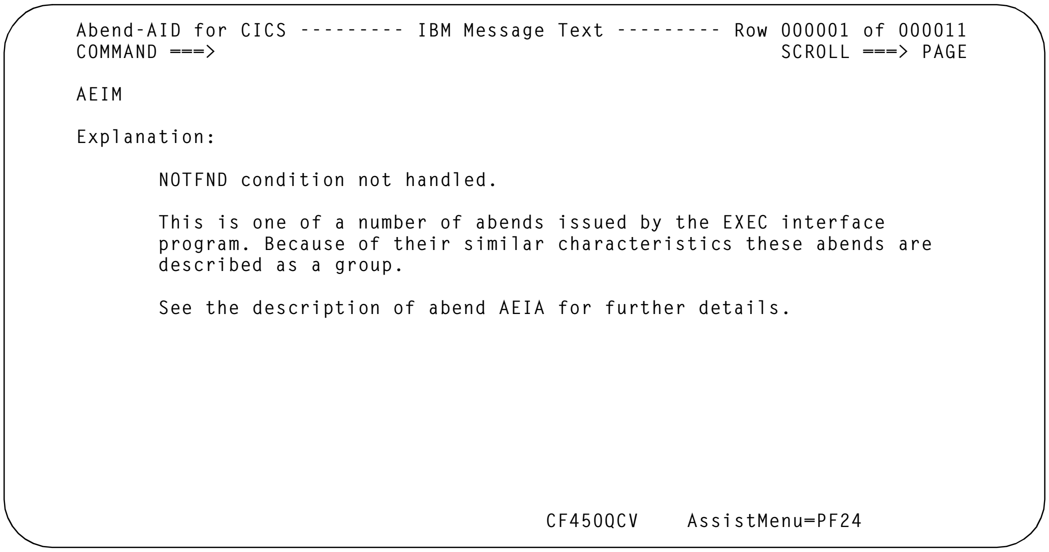


```
Abend-AID for CICS ------ Diagnostic Information ------ Row 000001 of 000009
COMMAND ===>                                              SCROLL ===> PAGE

The transaction was terminated with the AEIM abend because the excpetional
condition NOTFND occurred for which there was no 'EXEC CICS HANDLE
CONDITION' request active.

Either change the application program to prevent the condition recurring, or
to handle the condition using the "EXEC CICS HANDLE CONDITION' command.  If
necessary, use the contents of the EIBRCODE field in the EIB or information
from the CICS trace table to assist in determining the cause of the
exception condition.
```

Since an AEIM abend was the last abend to occur, this screen automatically displays the diagnostics for an AEIM. The diagnostics for all CICS abend codes can be displayed from this screen.

4. To display more information for an AEIM abend, tab to the highlighted AEIM abend code and press Enter. The IBM Message Text screen shown in Figure 9-11 on page 9-7 will be displayed.

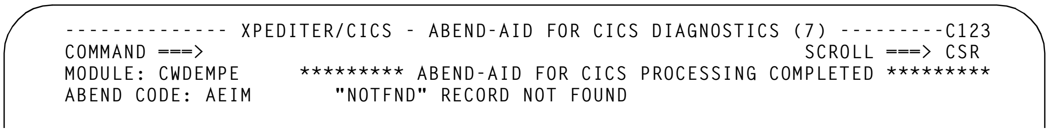
Figure 9-11. Abend-AID for CICS IBM Message Text Screen



```
Abend-AID for CICS --------- IBM Message Text --------- Row 000001 of 000011
COMMAND ===>                                                  SCROLL ===> PAGE

AEIM

Explanation:

       NOTFND condition not handled.

       This is one of a number of abends issued by the EXEC interface
       program. Because of their similar characteristics these abends are
       described as a group.

       See the description of abend AEIA for further details.







                                        CF450QCV    AssistMenu=PF24
```

5. Press PF3 twice to return to Xpediter/CICS. The Abend-AID for CICS Diagnostics screen (7) will be displayed as shown in Figure 9-12.

Figure 9-12. Abend-AID for CICS Diagnostics Screen (7)



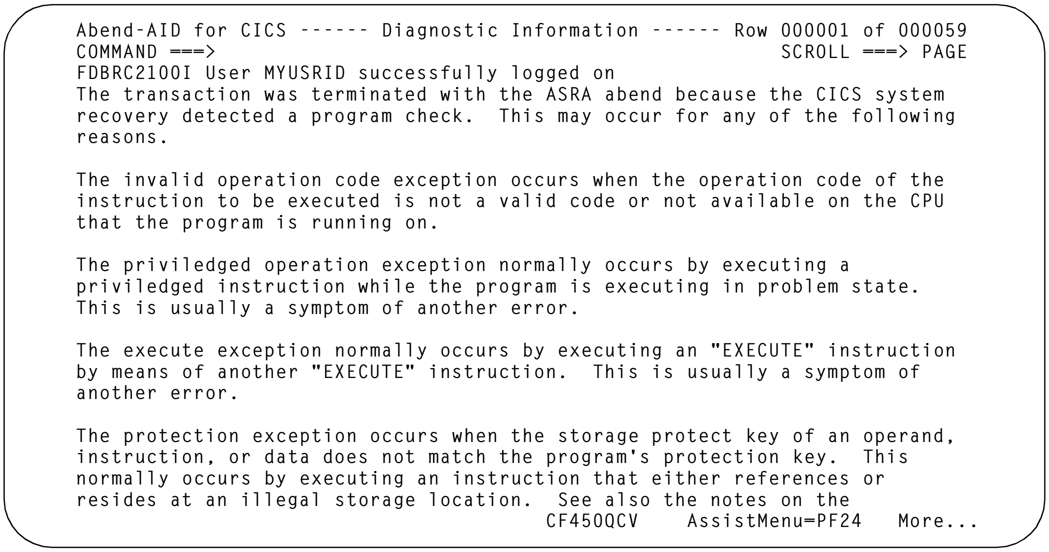
```
-------------- XPEDITER/CICS - ABEND-AID FOR CICS DIAGNOSTICS (7) ---------C123
COMMAND ===>                                                       SCROLL ===> CSR
MODULE: CWDEMPE     ********* ABEND-AID FOR CICS PROCESSING COMPLETED *********
ABEND CODE: AEIM       "NOTFND" RECORD NOT FOUND
```

6. To see another diagnostic message, type ASRA in the ABEND CODE field.

7. Press Enter. The Abend-AID for CICS Diagnostic Information screen appears as shown in Figure 9-13, with the explanation of an ASRA abend.

Remember to end the session as described in “Exiting Xpediter/CICS and Ending a Debugging Session” on page 2-6.

Figure 9-13. Abend-AID for CICS Diagnostic Information Screen for an ASRA Abend



```
Abend-AID for CICS ------ Diagnostic Information ------ Row 000001 of 000059
COMMAND ===>                                                 SCROLL ===> PAGE
FDBRC2100I User MYUSRID successfully logged on
The transaction was terminated with the ASRA abend because the CICS system
recovery detected a program check.  This may occur for any of the following
reasons.

The invalid operation code exception occurs when the operation code of the
instruction to be executed is not a valid code or not available on the CPU
that the program is running on.

The priviledged operation exception normally occurs by executing a
priviledged instruction while the program is executing in problem state.
This is usually a symptom of another error.

The execute exception normally occurs by executing an "EXECUTE" instruction
by means of another "EXECUTE" instruction.  This is usually a symptom of
another error.

The protection exception occurs when the storage protect key of an operand,
instruction, or data does not match the program's protection key.  This
normally occurs by executing an instruction that either references or
resides at an illegal storage location.  See also the notes on the
                                        CF450QCV    AssistMenu=PF24   More...
```

Chapter 10. Using Automatic Trap Activation

This chapter demonstrates Xpediter's Automatic Trap Activation (ATA) feature. ATA traps terminal-related transaction abends—but ***not*** breakpoints—without the user having an Xpediter session running.

Enabling the ATA feature is optional and is done with the ATA global table parameter. With the parameter set to OFF (the default), ATA is deactivated. Setting the parameter to XPED, XPRT, or XPSP activates Automatic Trap Activation.

With ATA activated, if a terminal-related transaction is about to abend, the abend will be trapped and Xpediter will be invoked on the terminal or 3270 Web Bridge session where the transaction was initiated. The transaction used to invoke Xpediter is determined by the value of the ATA global parameter. If the ATASCREEN global parameter is set to YES, a customizable notification/decision screen will first be displayed. If the user decides to debug the abending transaction, the Source Listing (2.L), Break/Abend (2.1), or Assembler Break/Abend (2.20) screen will be displayed, depending on the transaction specified for the ATA parameter and the availability of program source.

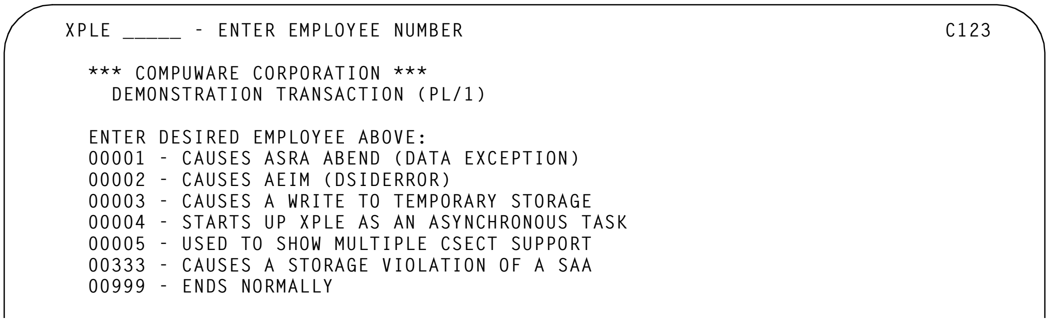
Trapping an Abend with ATA

In the following demonstration, the global parameter ATA has been set to XPED, ATASCREEN has been set to YES, and source is available for program CWDEMPE.

Note: For the purpose of properly demonstrating Automatic Trap Activation—but ***not*** for regular Xpediter/CICS use—you will first make sure Xpediter is not active on the terminal being used.

1. On a blank CICS screen, type **XPND** and press Enter. A message will be displayed saying either Xpediter is not active or it has been terminated.
2. On a blank CICS screen, type **XPLE** and press Enter. The Demonstration Transaction screen shown in Figure 10-1 will be displayed.

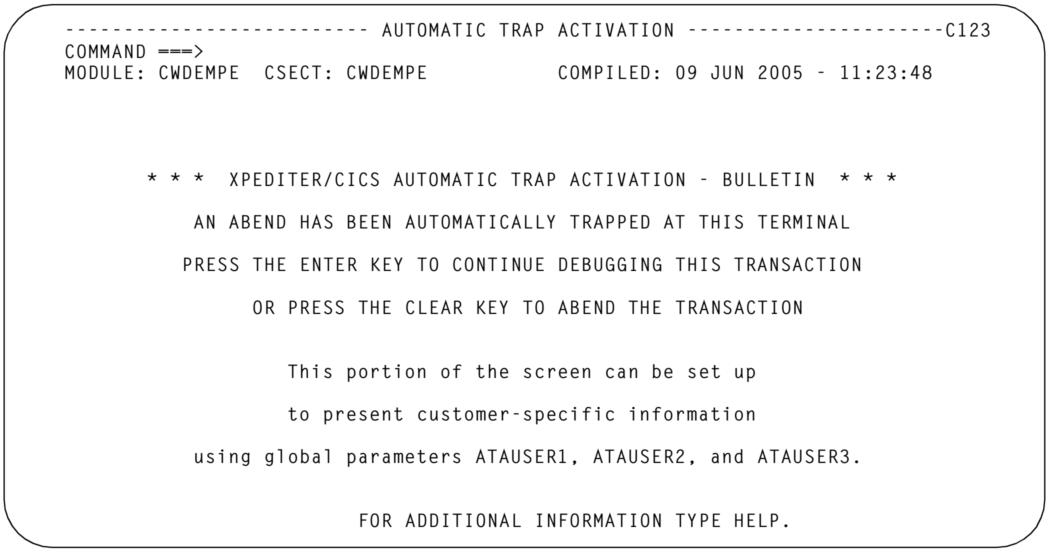
Figure 10-1. Demonstration Transaction Screen



```
XPLE _____ - ENTER EMPLOYEE NUMBER                                   C123

  *** COMPUWARE CORPORATION ***
    DEMONSTRATION TRANSACTION (PL/1)

  ENTER DESIRED EMPLOYEE ABOVE:
  00001 - CAUSES ASRA ABEND (DATA EXCEPTION)
  00002 - CAUSES AEIM (DSIDERROR)
  00003 - CAUSES A WRITE TO TEMPORARY STORAGE
  00004 - STARTS UP XPLE AS AN ASYNCHRONOUS TASK
  00005 - USED TO SHOW MULTIPLE CSECT SUPPORT
  00333 - CAUSES A STORAGE VIOLATION OF A SAA
  00999 - ENDS NORMALLY
```

3. To cause an ASRA abend, type **00001** and press Enter. The Automatic Trap Activation screen (Figure 10-2 on page 10-2) will be displayed. Pressing Clear will allow the transaction abend to occur, while pressing Enter will display the appropriate Xpediter screen for debugging the transaction.

Figure 10-2. Automatic Trap Activation Screen



```
-------------------------- AUTOMATIC TRAP ACTIVATION ----------------------C123
COMMAND ===>
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48



       * * *  XPEDITER/CICS AUTOMATIC TRAP ACTIVATION - BULLETIN  * * *

           AN ABEND HAS BEEN AUTOMATICALLY TRAPPED AT THIS TERMINAL

          PRESS THE ENTER KEY TO CONTINUE DEBUGGING THIS TRANSACTION

                OR PRESS THE CLEAR KEY TO ABEND THE TRANSACTION

                   This portion of the screen can be set up

                   to present customer-specific information

           using global parameters ATAUSER1, ATAUSER2, and ATAUSER3.

                         FOR ADDITIONAL INFORMATION TYPE HELP.
```

4. Press Enter. The Source Listing screen (2.L) will be displayed with the source of program CWDEMPE positioned to the instruction at which the abend occurred.

At this point, the user would debug the abending transaction as described in "Testing without Breakpoints" on page 3-1.

Chapter 11.
Setting Up a Profile

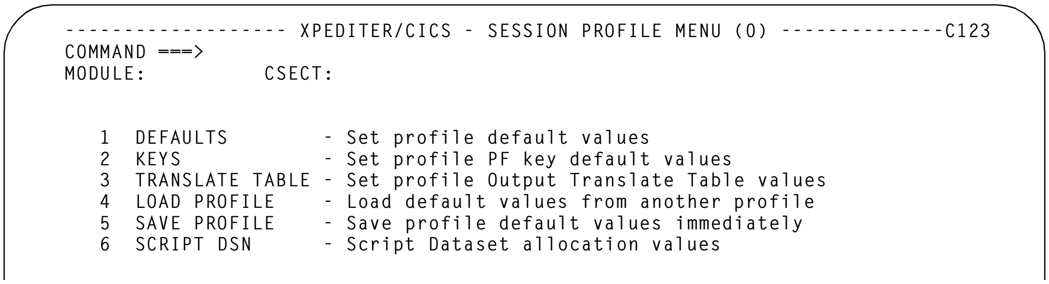
Xpediter/CICS allows you to have an individual user profile to customize your debugging session for your needs and preferences. A profile is a set of default values that you have specified and stored for your own use. This chapter discusses how to set up your own profile.

When you use Xpediter/CICS, the system will recognize your userid and will call up your profile. Then when you work with fields and actions that have defaults, Xpediter/CICS will use defaults from your profile instead of those furnished by Xpediter itself.

1. Start Xpediter/CICS by entering **XPED** on a blank CICS screen.
2. To access the Session Profile Menu (0), type **0** in the COMMAND field of the Primary Menu.
3. Press Enter to display the Session Profile Menu (0) as shown in Figure 11-1.

Note: Menu option 6 SCRIPT DSN is not displayed if global parameter XDSCRPT is set to NO. The default is YES.

Figure 11-1. Session Profile Menu (0)



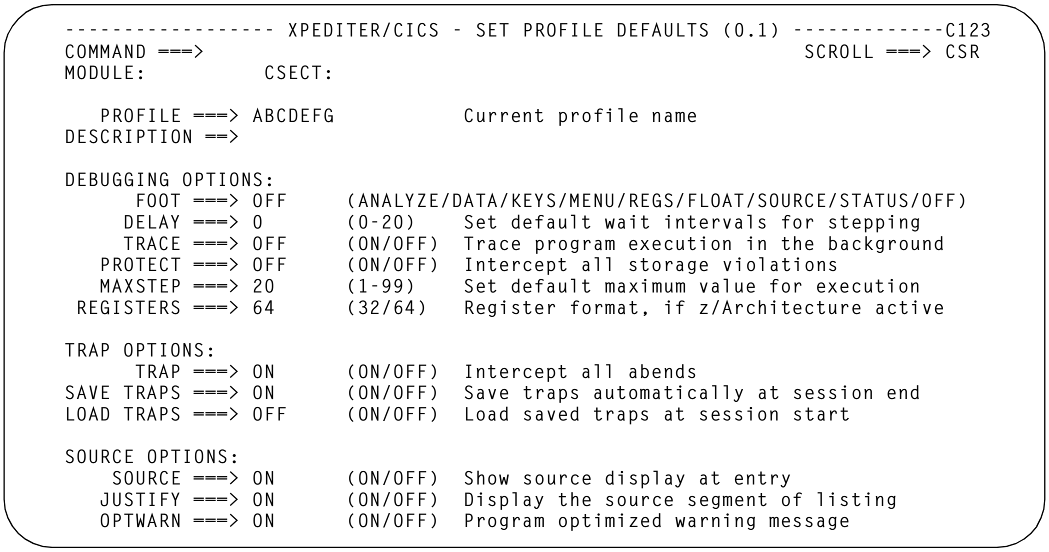
```
------------------ XPEDITER/CICS - SESSION PROFILE MENU (0) -------------C123
COMMAND ===>
MODULE:          CSECT:


   1  DEFAULTS        - Set profile default values
   2  KEYS            - Set profile PF key default values
   3  TRANSLATE TABLE - Set profile Output Translate Table values
   4  LOAD PROFILE    - Load default values from another profile
   5  SAVE PROFILE    - Save profile default values immediately
   6  SCRIPT DSN      - Script Dataset allocation values
```

Changing Profile Defaults

1. To access the Set Profile Defaults screen (0.1), type **=0.1** in the COMMAND field of any Xpediter/CICS screen.
2. Press Enter to display the Set Profile Defaults screen (0.1) (Figure 11-2). The Set Profile Defaults screen (0.1) controls environmental features, such as scroll values and the type of information that Xpediter/CICS displays on the bottom of the screen. This is a two-page screen. The second page is accessed by scrolling forward (PF8) (Figure 11-3).

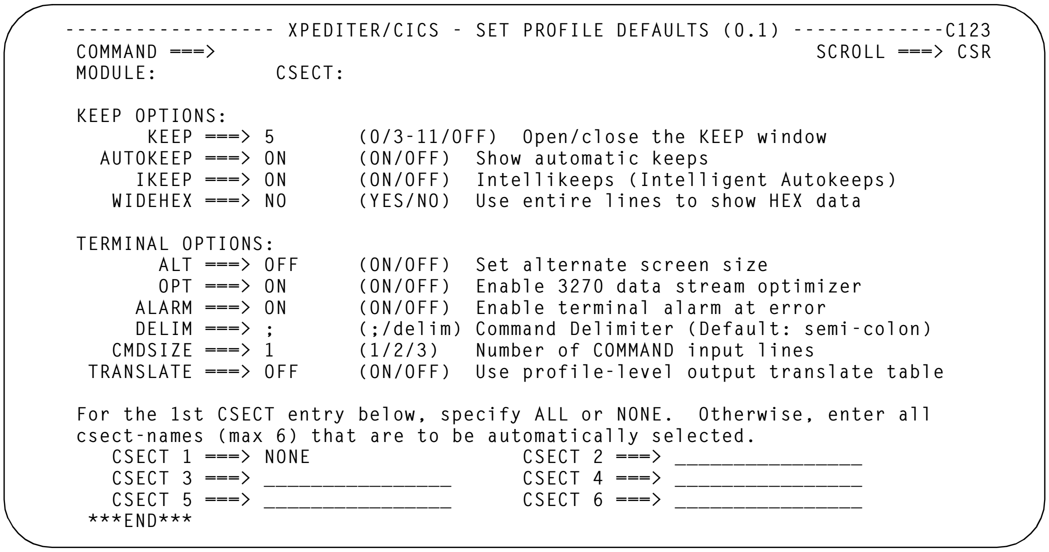
Note: Certain settings may be overridden or non-applicable when using Xpediter in one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

Figure 11-2. Set Profile Defaults Screen 1



```
----------------- XPEDITER/CICS - SET PROFILE DEFAULTS (0.1) -------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE:          CSECT:

   PROFILE ===> ABCDEFG           Current profile name
DESCRIPTION ==>

DEBUGGING OPTIONS:
      FOOT ===> OFF     (ANALYZE/DATA/KEYS/MENU/REGS/FLOAT/SOURCE/STATUS/OFF)
     DELAY ===> 0       (0-20)    Set default wait intervals for stepping
     TRACE ===> OFF     (ON/OFF)  Trace program execution in the background
   PROTECT ===> OFF     (ON/OFF)  Intercept all storage violations
   MAXSTEP ===> 20      (1-99)    Set default maximum value for execution
 REGISTERS ===> 64      (32/64)   Register format, if z/Architecture active

TRAP OPTIONS:
      TRAP ===> ON      (ON/OFF)  Intercept all abends
SAVE TRAPS ===> ON      (ON/OFF)  Save traps automatically at session end
LOAD TRAPS ===> OFF     (ON/OFF)  Load saved traps at session start

SOURCE OPTIONS:
    SOURCE ===> ON      (ON/OFF)  Show source display at entry
   JUSTIFY ===> ON      (ON/OFF)  Display the source segment of listing
   OPTWARN ===> ON      (ON/OFF)  Program optimized warning message
```

Figure 11-3. Set Profile Defaults Screen 2



```
----------------- XPEDITER/CICS - SET PROFILE DEFAULTS (0.1) -------------C123
 COMMAND ===>                                                  SCROLL ===> CSR
 MODULE:          CSECT:

 KEEP OPTIONS:
       KEEP ===> 5       (0/3-11/OFF)  Open/close the KEEP window
   AUTOKEEP ===> ON      (ON/OFF)  Show automatic keeps
      IKEEP ===> ON      (ON/OFF)  Intellikeeps (Intelligent Autokeeps)
    WIDEHEX ===> NO      (YES/NO)  Use entire lines to show HEX data

 TERMINAL OPTIONS:
        ALT ===> OFF     (ON/OFF)  Set alternate screen size
        OPT ===> ON      (ON/OFF)  Enable 3270 data stream optimizer
      ALARM ===> ON      (ON/OFF)  Enable terminal alarm at error
      DELIM ===> ;       (;/delim) Command Delimiter (Default: semi-colon)
    CMDSIZE ===> 1       (1/2/3)   Number of COMMAND input lines
  TRANSLATE ===> OFF     (ON/OFF)  Use profile-level output translate table

 For the 1st CSECT entry below, specify ALL or NONE.  Otherwise, enter all
 csect-names (max 6) that are to be automatically selected.
    CSECT 1 ===> NONE                  CSECT 2 ===> ________________
    CSECT 3 ===> ________________      CSECT 4 ===> ________________
    CSECT 5 ===> ________________      CSECT 6 ===> ________________
  ***END***
```

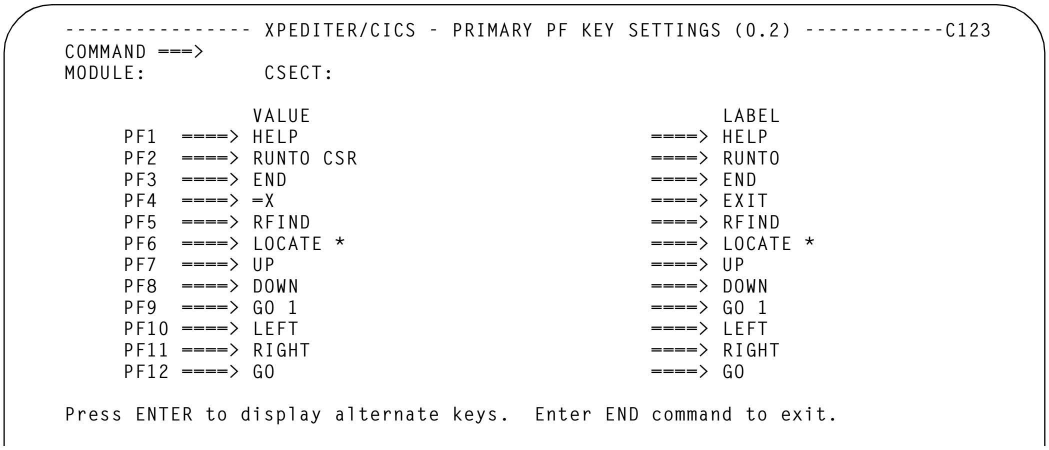
3. To change a default option, overtype the current value with the new value in the appropriate field.
4. Press Enter. The Set Profile Defaults screen (0.1) is updated to show the new value.

Changing PF Key Settings

You can change the PF key functions and labels to suit your needs.

1. To access the PF key setting screen (0.2), type **=0.2** in the COMMAND field of any Xpediter/CICS screen.
2. Press Enter to display the Primary PF Key Settings screen (0.2) (Figure 11-4). On this screen you can modify settings for PF1 through PF12.

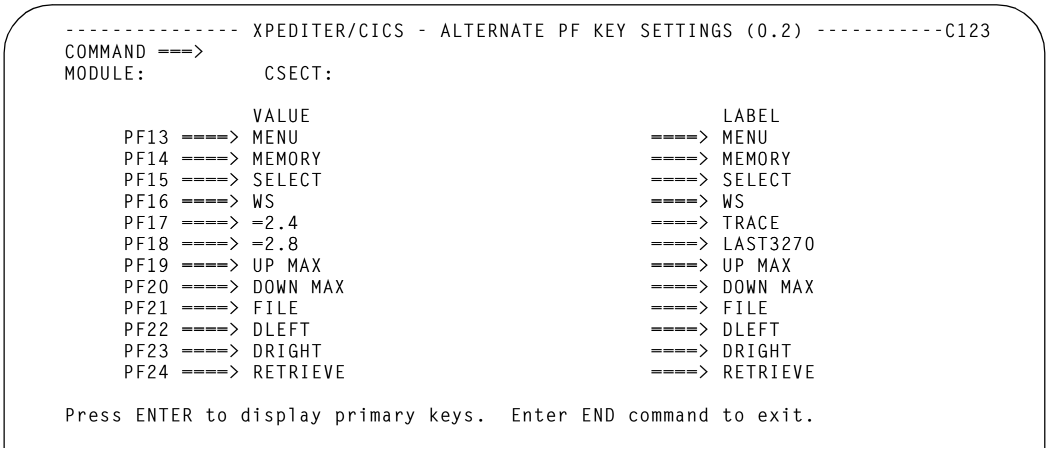
Figure 11-4. Primary PF Key Settings Screen (0.2)



```
--------------- XPEDITER/CICS - PRIMARY PF KEY SETTINGS (0.2) ------------C123
COMMAND ===>
MODULE:          CSECT:

                VALUE                                   LABEL
     PF1  ====> HELP                               ====> HELP
     PF2  ====> RUNTO CSR                          ====> RUNTO
     PF3  ====> END                                ====> END
     PF4  ====> =X                                 ====> EXIT
     PF5  ====> RFIND                              ====> RFIND
     PF6  ====> LOCATE *                           ====> LOCATE *
     PF7  ====> UP                                 ====> UP
     PF8  ====> DOWN                               ====> DOWN
     PF9  ====> GO 1                               ====> GO 1
     PF10 ====> LEFT                               ====> LEFT
     PF11 ====> RIGHT                              ====> RIGHT
     PF12 ====> GO                                 ====> GO

Press ENTER to display alternate keys.  Enter END command to exit.
```

3. Press Enter to update PF1 through PF12 and display PF13 through PF24. The Alternate PF Key Settings screen (0.2) appears as shown in Figure 11-5. On this screen you can modify settings for PF13 through PF24.
4. Press Enter to update PF13 through PF24 and redisplay the Primary PF Key Settings screen (0.2).
5. To change the function of PF9, type **GO 5** in the VALUE field next to PF9.
6. To change the label for PF9, type **GO 5** in the LABEL field next to PF9. Press Enter.

Figure 11-5. Alternate PF Key Settings Screen (0.2)



```
-------------- XPEDITER/CICS - ALTERNATE PF KEY SETTINGS (0.2) -----------C123
COMMAND ===>
MODULE:          CSECT:

                VALUE                                   LABEL
     PF13 ====> MENU                               ====> MENU
     PF14 ====> MEMORY                             ====> MEMORY
     PF15 ====> SELECT                             ====> SELECT
     PF16 ====> WS                                 ====> WS
     PF17 ====> =2.4                               ====> TRACE
     PF18 ====> =2.8                               ====> LAST3270
     PF19 ====> UP MAX                             ====> UP MAX
     PF20 ====> DOWN MAX                           ====> DOWN MAX
     PF21 ====> FILE                               ====> FILE
     PF22 ====> DLEFT                              ====> DLEFT
     PF23 ====> DRIGHT                             ====> DRIGHT
     PF24 ====> RETRIEVE                           ====> RETRIEVE

Press ENTER to display primary keys.  Enter END command to exit.
```

The default PF key values can also be changed using the KEYS primary command from any screen.

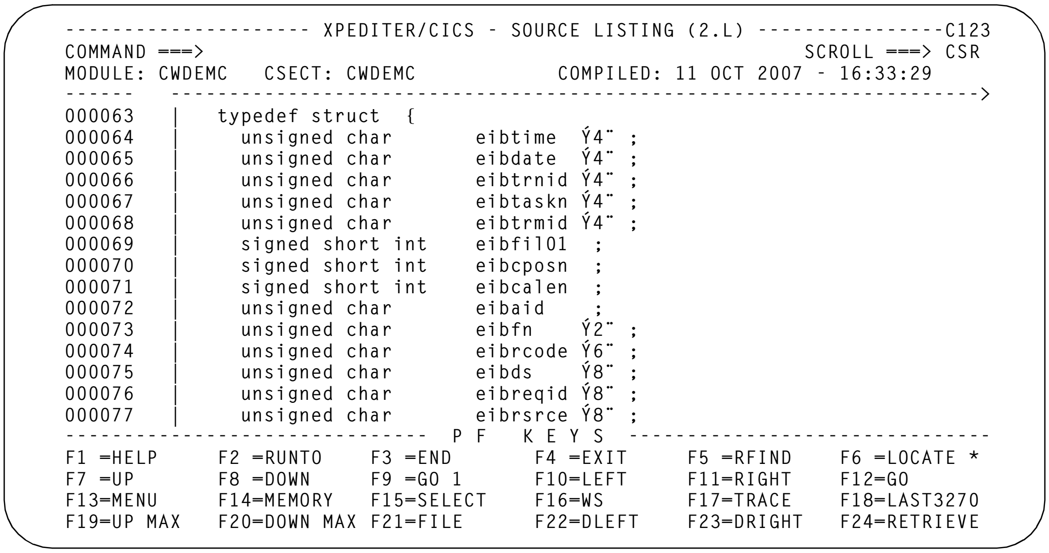
Note: The values in the LABEL column also appear on the PF key buttons displayed when using Xpediter's 3270 Web Bridge support.

Changing Output Translate Table Values

The Output Translate Table can be used to allow the display of non-English characters when your terminal's codepage is not supported by the Xpediter/CICS global NLS parameter. It is also useful when there is a mismatch between a listing's codepage and your terminal codepage. The following example shows how C-language programmers can display open and close brackets within a listing that was compiled with codepage 1047, when their terminal codepage is 037. We will change the listing's bracket positions in this table (x'AD' and x'BD' for IBM-1047 codepage) to the bracket values of your terminal's codepage (x'BA' and x'BB', for IBM-037 codepage, respectively).

1. Start with a display of our C-language demo program CWDEMC on the 2.L screen. Type **UP MAX** on the command line and press Enter. Type **FIND eibtime** on the command line, and press Enter. Eibtime has left and right brackets on it. Note that those brackets don't display as brackets on your terminal.

Figure 11-6. Xpediter/CICS - Source Listing Screen (2.L)

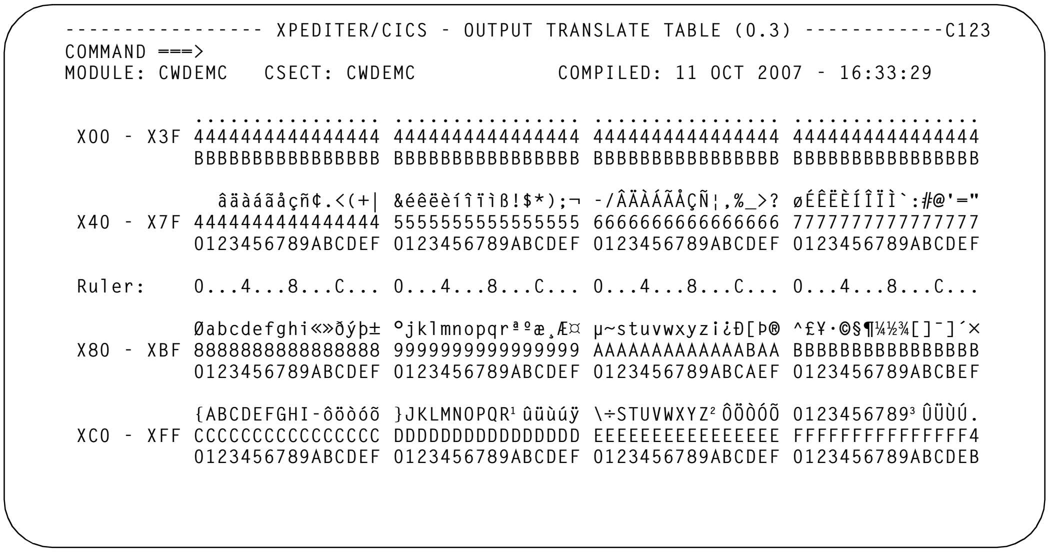


```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMC   CSECT: CWDEMC            COMPILED: 11 OCT 2007 - 16:33:29
------   ------------------------------------------------------------------->
000063   |   typedef struct  {
000065   |     unsigned char       eibdate  Ý4¨ ;
000064   |     unsigned char       eibtime  Ý4¨ ;
000066   |     unsigned char       eibtrnid Ý4¨ ;
000067   |     unsigned char       eibtaskn Ý4¨ ;
000068   |     unsigned char       eibtrmid Ý4¨ ;
000069   |     signed short int    eibfil01  ;
000070   |     signed short int    eibcposn  ;
000071   |     signed short int    eibcalen  ;
000072   |     unsigned char       eibaid    ;
000073   |     unsigned char       eibfn    Ý2¨ ;
000074   |     unsigned char       eibrcode Ý6¨ ;
000075   |     unsigned char       eibds    Ý8¨ ;
000076   |     unsigned char       eibreqid Ý8¨ ;
000077   |     unsigned char       eibrsrce Ý8¨ ;
-------------------------------- P F   K E Y S -------------------------------
F1 =HELP     F2 =RUNTO    F3 =END       F4 =EXIT     F5 =RFIND    F6 =LOCATE *
F7 =UP       F8 =DOWN     F9 =GO 1      F10=LEFT     F11=RIGHT    F12=GO
F13=MENU     F14=MEMORY   F15=SELECT    F16=WS       F17=TRACE    F18=LAST3270
F19=UP MAX   F20=DOWN MAX F21=FILE      F22=DLEFT    F23=DRIGHT   F24=RETRIEVE
```

2. Type **=0.3** on the command line to go to the Output Translate Table. Go to the "AD" vertical hex spot found on the X80-XBF line, overtype the A (found on the top line) with a B, and the D (on the bottom line) with an A and press Enter. You have now replaced the X'AD' with X'BA'.

3. Go to the "BD" vertical hex spot also found on the X80-XBF line and leave the B (top line) as is, and overtype the D (bottom line) with a **B** and press Enter. You have now replaced the X'BD' with X'BB'.

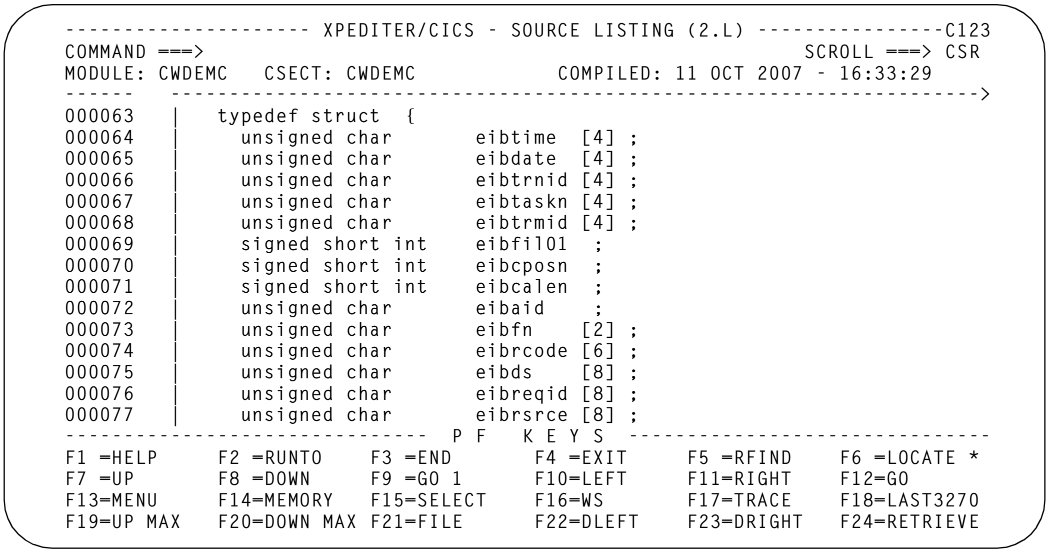
Figure 11-7. Xpediter/CICS - Output Translate Table Screen (0.3) after changing X'AD' and X'BD'



```
 ---------------- XPEDITER/CICS - OUTPUT TRANSLATE TABLE (0.3) -----------C123
 COMMAND ===>
 MODULE: CWDEMC   CSECT: CWDEMC           COMPILED: 11 OCT 2007 - 16:33:29

           ................ ................ ................ ................
  X00 - X3F 4444444444444444 4444444444444444 4444444444444444 4444444444444444
           BBBBBBBBBBBBBBBB BBBBBBBBBBBBBBBB BBBBBBBBBBBBBBBB BBBBBBBBBBBBBBBB

             âäàáãåçñ¢.<(+| &éêëèíîïìß!$*);¬ -/ÂÄÀÁÃÅÇÑ¦,%_>? øÉÊËÈÍÎÏÌ`:#@'="
  X40 - X7F 4444444444444444 5555555555555555 6666666666666666 7777777777777777
           0123456789ABCDEF 0123456789ABCDEF 0123456789ABCDEF 0123456789ABCDEF

  Ruler:    0...4...8...C... 0...4...8...C... 0...4...8...C... 0...4...8...C...

           Øabcdefghi«»ðýþ± °jklmnopqrªºæ¸Æ¤ µ~stuvwxyz¡¿Ð[Þ® ^£¥·©§¶¼½¾[]¯]´×
  X80 - XBF 8888888888888888 9999999999999999 AAAAAAAAAAAAABAA BBBBBBBBBBBBBBBB
           0123456789ABCDEF 0123456789ABCDEF 0123456789ABCAEF 0123456789ABCBEF

           {ABCDEFGHI-ôöòóõ }JKLMNOPQR¹ûüùúÿ \÷STUVWXYZ²ÔÖÒÓÕ 0123456789³ÛÜÙÚ.
  XC0 - XFF CCCCCCCCCCCCCCCC DDDDDDDDDDDDDDDD EEEEEEEEEEEEEEEE FFFFFFFFFFFFFFF4
           0123456789ABCDEF 0123456789ABCDEF 0123456789ABCDEF 0123456789ABCDEB
```

4. Return to the Source Listing screen (2.L) to see your changes. If you aren't still positioned on the eibtime field, repeat the UP MAX and FIND eibtime commands. You should now see the brackets display as brackets.

Figure 11-8. Xpediter/CICS - Source Listing Screen (2.L) with display brackets



```
 -------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ---------------C123
 COMMAND ===>                                                 SCROLL ===> CSR
 MODULE: CWDEMC   CSECT: CWDEMC           COMPILED: 11 OCT 2007 - 16:33:29
 ------    ------------------------------------------------------------------->
 000063  |   typedef struct  {
 000064  |     unsigned char       eibtime  [4] ;
 000065  |     unsigned char       eibdate  [4] ;
 000066  |     unsigned char       eibtrnid [4] ;
 000067  |     unsigned char       eibtaskn [4] ;
 000068  |     unsigned char       eibtrmid [4] ;
 000069  |     signed short int    eibfil01  ;
 000070  |     signed short int    eibcposn  ;
 000071  |     signed short int    eibcalen  ;
 000072  |     unsigned char       eibaid    ;
 000073  |     unsigned char       eibfn    [2] ;
 000074  |     unsigned char       eibrcode [6] ;
 000075  |     unsigned char       eibds    [8] ;
 000076  |     unsigned char       eibreqid [8] ;
 000077  |     unsigned char       eibrsrce [8] ;
 ------------------------------  P F   K E Y S  ------------------------------
 F1 =HELP     F2 =RUNTO    F3 =END       F4 =EXIT     F5 =RFIND    F6 =LOCATE *
 F7 =UP       F8 =DOWN     F9 =GO 1      F10=LEFT     F11=RIGHT    F12=GO
 F13=MENU     F14=MEMORY   F15=SELECT    F16=WS       F17=TRACE    F18=LAST3270
 F19=UP MAX   F20=DOWN MAX F21=FILE      F22=DLEFT    F23=DRIGHT   F24=RETRIEVE
```

Loading a Profile

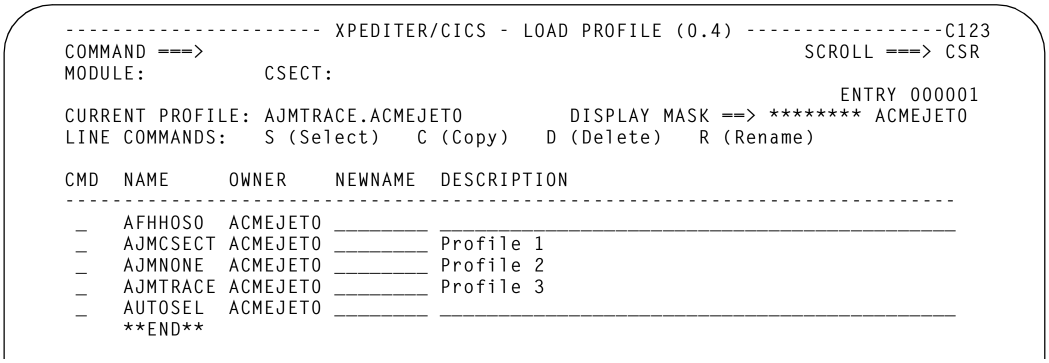
Once a profile is created, it can be loaded whenever XPED is entered. To load a profile, enter **XPED P =** ***profile*** from a blank CICS screen, where *profile* equals the profile name. For example, to load ALTKEYS, enter **XPED P=ALTKEYS**.

An alternate profile can be loaded at any time by accessing the Load Profile screen (0.4) (Figure 11-9).

The current profile name can be changed via the SELECT line command on the Load Profile (0.4) screen or by accessing the Set Profile Defaults screen (0.1) and over-typing the profile name.

Note: If user ID is used to sign onto the CICS region, a profile for the user ID is automatically created. If the P= option for specifying a profile is not used when the Xpediter/CICS session is initiated, the profile for the user ID is automatically loaded and used.

Figure 11-9. Load Profile Screen (0.4)

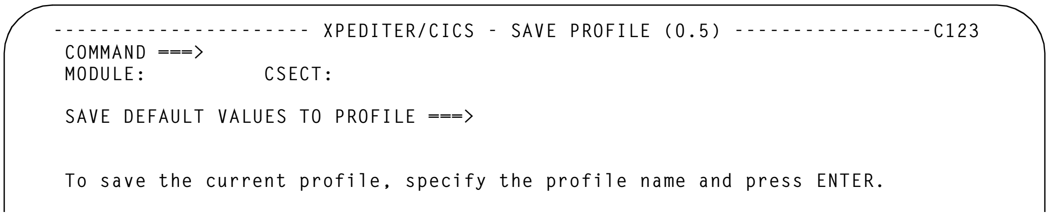


```
--------------------- XPEDITER/CICS - LOAD PROFILE (0.4) -----------------C123
COMMAND ===>                                                 SCROLL ===> CSR
MODULE:          CSECT:
                                                                ENTRY 000001
CURRENT PROFILE: AJMTRACE.ACMEJETO         DISPLAY MASK ==> ******** ACMEJETO
LINE COMMANDS:   S (Select)   C (Copy)   D (Delete)   R (Rename)

CMD  NAME     OWNER    NEWNAME  DESCRIPTION
----------------------------------------------------------------------------
 _   AFHHOSO  ACMEJETO ________ ____________________________________________
 _   AJMCSECT ACMEJETO ________ Profile 1
 _   AJMNONE  ACMEJETO ________ Profile 2
 _   AJMTRACE ACMEJETO ________ Profile 3
 _   AUTOSEL  ACMEJETO ________ ____________________________________________
     **END**
```

Saving Profile Defaults

1. To save the new profile, type **=0.5** in the COMMAND field and press Enter to display the Save Profile screen (0.5) (Figure 11-10).

Figure 11-10. Save Profile Screen (0.5)



```
--------------------- XPEDITER/CICS - SAVE PROFILE (0.5) -----------------C123
COMMAND ===>
MODULE:          CSECT:

SAVE DEFAULT VALUES TO PROFILE ===>


To save the current profile, specify the profile name and press ENTER.
```

2. Type **ALTKEYS** (or any new profile name) in the PROFILE NAME field.
3. Press Enter. Xpediter/CICS displays the message:

   ```
   *********** PROFILE HAS BEEN UPDATED ***********
   ```

   next to the MODULE field to show that the new profile has been saved.

Chapter 12.
Accessing Files

This chapter introduces the Xpediter/CICS file utility, which displays lists of resources that you can access under CICS, including VSAM and BDAM datasets, DL/1 databases, DB2 data, transient data, temporary storage queues, and MQ queues. Choose the resource you want to access from these lists and perform any of the displayed service requests.

Once a resource is selected, Xpediter/CICS displays the records in that resource. You can then add, delete, or modify the records. Security is available to restrict update and delete access. To provide audit capabilities, Xpediter's Log Facility gives sites the option of logging all changes made to supported resources. For more information, refer to the *Xpediter/CICS Installation Guide*.

This chapter shows how to browse records in a file and change file service requests. During these exercises, you will resolve two abends, AEIM and AEIP, that can occur when working with VSAM files.

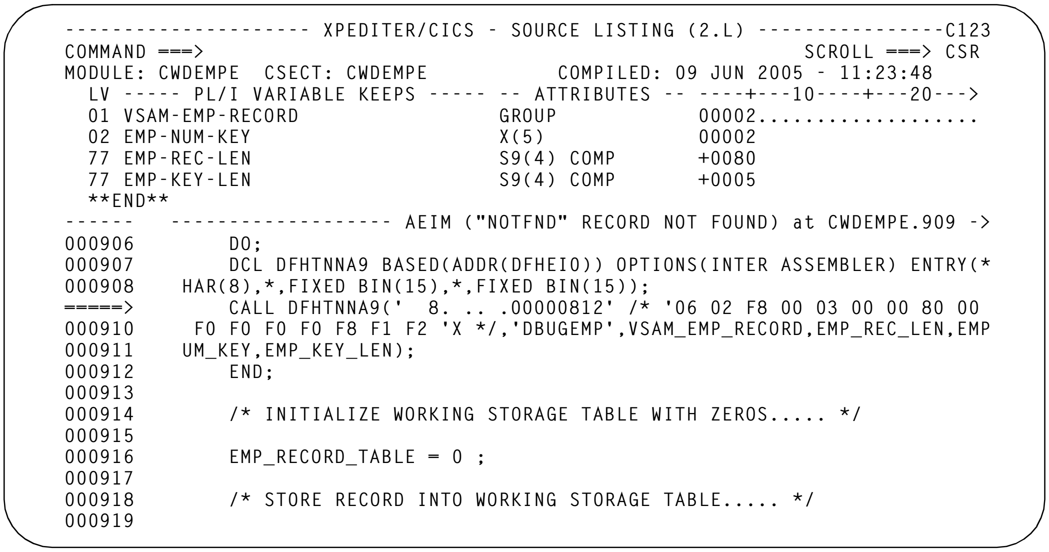
Note: The demonstrations in this chapter should be performed in Xpediter's standard operating mode — ***not*** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

Browsing Records in a File

1. Start Xpediter/CICS by entering **XPED** on a blank CICS screen and pressing Enter.
2. Press Clear to return to CICS.
3. Start the transaction by typing XPLE in the top left corner of a blank CICS screen.
4. Press Enter to display the Demonstration Transaction screen.
5. Type **00002** to cause an AEIM abend.
6. Press Enter. Xpediter/CICS intercepts an AEIM abend and displays the Source Listing screen (2.L) (Figure 12-1).

   Look at the data in the keep window. This program is attempting to find record 00002 (EMP_NUM_KEY) in the DBUGEMP file. You can check the DBUGEMP file to see if that record is in the file.

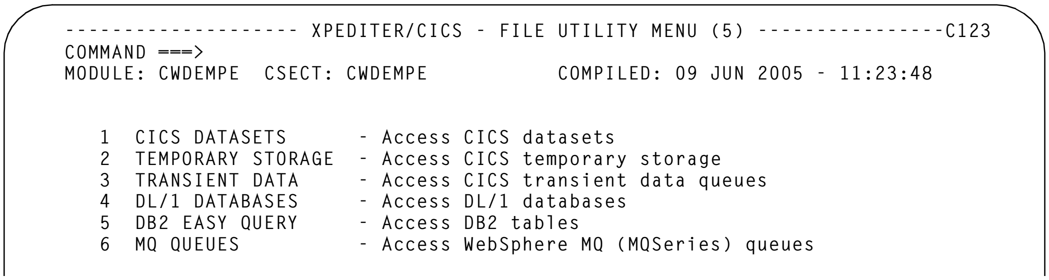
Figure 12-1. Record Not Found Message on the Source Listing Screen (2.L)



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ---------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
  LV ----- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
  01 VSAM-EMP-RECORD                 GROUP            00002..................
  02 EMP-NUM-KEY                     X(5)             00002
  77 EMP-REC-LEN                     S9(4) COMP       +0080
  77 EMP-KEY-LEN                     S9(4) COMP       +0005
  **END**
------   ------------------- AEIM ("NOTFND" RECORD NOT FOUND) at CWDEMPE.909 ->
000906       DO;
000907       DCL DFHTNNA9 BASED(ADDR(DFHEI0)) OPTIONS(INTER ASSEMBLER) ENTRY(*
000908    HAR(8),*,FIXED BIN(15),*,FIXED BIN(15));
=====>       CALL DFHTNNA9('  8. .. .00000812' /* '06 02 F8 00 03 00 00 80 00
000910     F0 F0 F0 F0 F8 F1 F2 'X */,'DBUGEMP',VSAM_EMP_RECORD,EMP_REC_LEN,EMP
000911    UM_KEY,EMP_KEY_LEN);
000912       END;
000913
000914       /* INITIALIZE WORKING STORAGE TABLE WITH ZEROS..... */
000915
000916       EMP_RECORD_TABLE = 0 ;
000917
000918       /* STORE RECORD INTO WORKING STORAGE TABLE..... */
000919
```

7. Type **FILE** in the COMMAND field to transfer to the File Utility.
8. Press Enter to display the File Utility Menu (5) (Figure 12-2).

Figure 12-2. File Utility Menu (5)

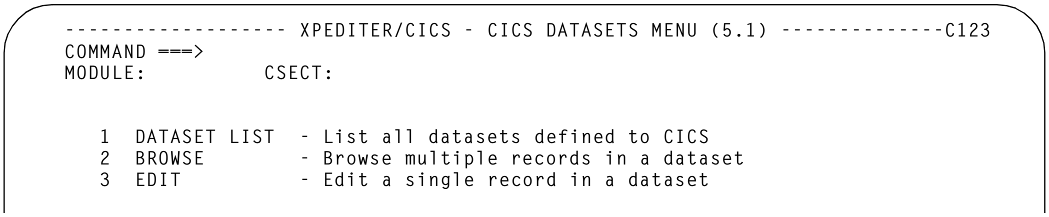


```
------------------- XPEDITER/CICS - FILE UTILITY MENU (5) ---------------C123
COMMAND ===>
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48


   1  CICS DATASETS      - Access CICS datasets
   2  TEMPORARY STORAGE  - Access CICS temporary storage
   3  TRANSIENT DATA     - Access CICS transient data queues
   4  DL/1 DATABASES     - Access DL/1 databases
   5  DB2 EASY QUERY     - Access DB2 tables
   6  MQ QUEUES          - Access WebSphere MQ (MQSeries) queues
```

The File Utility Menu (5) lists the types of resources that can be accessed through the file utility. Because the DBUGEMP dataset (the demonstration employee file) is a VSAM file, you will access it with the CICS DATASETS option.

9. Type **1** in the COMMAND field.
10. Press Enter to display the CICS Datasets Menu (5.1) (Figure 12-3), which lists the functions that you can perform on a dataset.

Figure 12-3. CICS Dataset Menu (5.1)

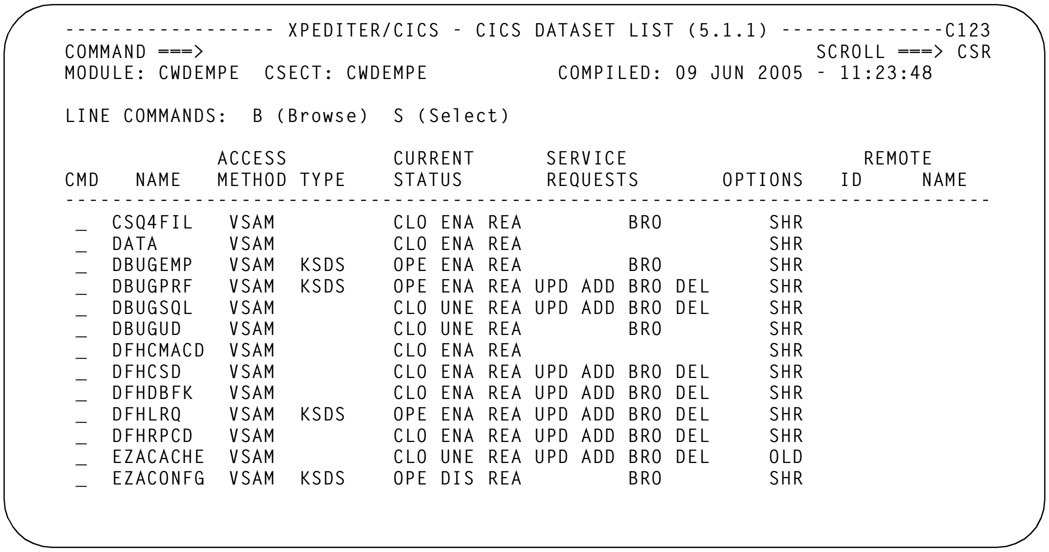


```
------------------ XPEDITER/CICS - CICS DATASETS MENU (5.1) -------------C123
COMMAND ===>
MODULE:          CSECT:


   1  DATASET LIST  - List all datasets defined to CICS
   2  BROWSE        - Browse multiple records in a dataset
   3  EDIT          - Edit a single record in a dataset
```

11. Type **1** in the COMMAND field to display a list of available datasets.
12. Press Enter to display the CICS Dataset List screen (5.1.1) (Figure 12-4).

    Note: The datasets displayed on your screen differ from the ones shown in this example. Xpediter/CICS lists all of the files in the file resources defined for your CICS region.

Figure 12-4. CICS Dataset List Screen (5.1.1)



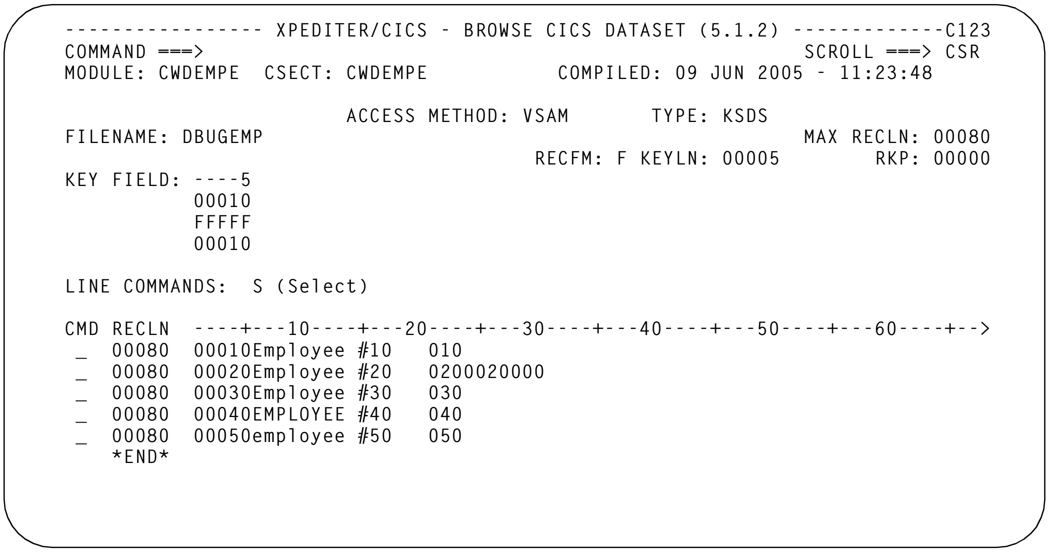
```
------------------ XPEDITER/CICS - CICS DATASET LIST (5.1.1) -------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48

LINE COMMANDS:  B (Browse)  S (Select)

             ACCESS         CURRENT      SERVICE                     REMOTE
CMD   NAME   METHOD TYPE    STATUS       REQUESTS       OPTIONS   ID     NAME
-------------------------------------------------------------------------------
 _  CSQ4FIL   VSAM          CLO ENA REA         BRO         SHR
 _  DATA      VSAM          CLO ENA REA                     SHR
 _  DBUGEMP   VSAM  KSDS    OPE ENA REA         BRO         SHR
 _  DBUGPRF   VSAM  KSDS    OPE ENA REA UPD ADD BRO DEL     SHR
 _  DBUGSQL   VSAM          CLO UNE REA UPD ADD BRO DEL     SHR
 _  DBUGUD    VSAM          CLO UNE REA         BRO         SHR
 _  DFHCMACD  VSAM          CLO ENA REA                     SHR
 _  DFHCSD    VSAM          CLO ENA REA UPD ADD BRO DEL     SHR
 _  DFHDBFK   VSAM          CLO ENA REA UPD ADD BRO DEL     SHR
 _  DFHLRQ    VSAM  KSDS    OPE ENA REA UPD ADD BRO DEL     SHR
 _  DFHRPCD   VSAM          CLO ENA REA UPD ADD BRO DEL     SHR
 _  EZACACHE  VSAM          CLO UNE REA UPD ADD BRO DEL     OLD
 _  EZACONFG  VSAM  KSDS    OPE DIS REA         BRO         SHR
```

The CICS Dataset List screen (5.1.1) lists the datasets that you can access from this CICS region as defined in the file resources. In addition to the dataset name, the screen displays the access method, current status, and allowed service requests.

Use PF7 and PF8 to scroll up and down through the list. Use the LOCATE command to find a particular dataset. Valid line commands are B (Browse) and S (Select). The B command displays a list of the records in the dataset. The S command displays the Edit CICS Dataset Record screen (5.1.3), where you can enter a record key.

13. Type **B** to the left of the DBUGEMP dataset and press Enter to display the Browse CICS Dataset screen (5.1.2) (Figure 12-5).

Figure 12-5. Browse CICS Dataset Screen (5.1.2)



```
----------------- XPEDITER/CICS - BROWSE CICS DATASET (5.1.2) -------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48

                        ACCESS METHOD: VSAM       TYPE: KSDS
FILENAME: DBUGEMP                                              MAX RECLN: 00080
                                        RECFM: F KEYLN: 00005        RKP: 00000
KEY FIELD: ----5
           00010
           FFFFF
           00010

LINE COMMANDS:  S (Select)

CMD RECLN  ----+---10----+---20----+---30----+---40----+---50----+---60----+-->
 _  00080  00010Employee #10   010
 _  00080  00020Employee #20   0200020000
 _  00080  00030Employee #30   030
 _  00080  00040EMPLOYEE #40   040
 _  00080  00050employee #50   050
    *END*
```

This screen displays a list of all records in the selected dataset. You can display the records in hexadecimal or character format. You can also scroll through the list and use the FIND command to find a particular string.

File information is shown at the top of the screen. The example shows that DBUGEMP is a key sequenced dataset (KSDS), with fixed records (RECFM: F) that have a key length (KEYLN) of 5. The relative key position (RKP) is 0, with a maximum record length (MAX RECLN) of 80.

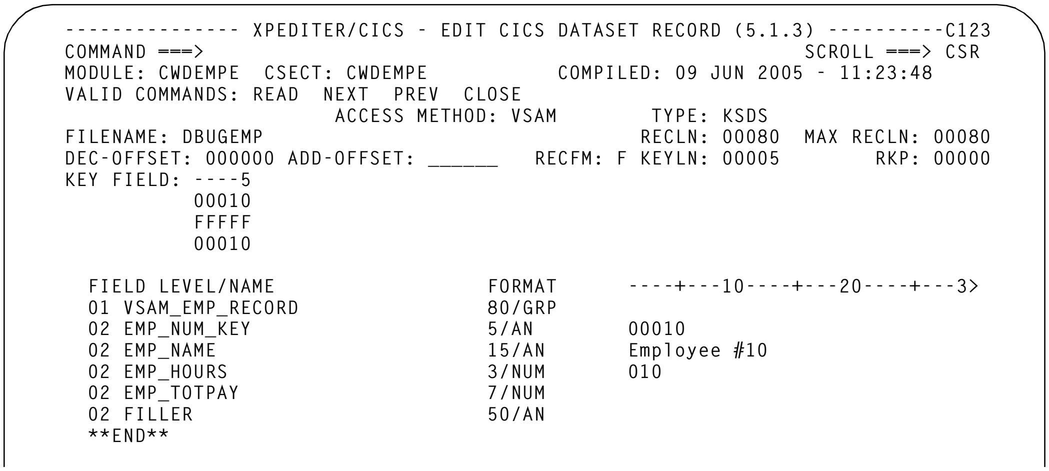
The KEY FIELD area shows the key of the record positioned at the top of the list. To position to another record, enter its key in the KEY FIELD.

The record information is displayed in character format. You can select individual records for update by typing an **S** next to the record and pressing Enter. You can also map records to a PLI copybook or data structure to display the field values next to their data names.

14. Type **S** to the left of the first record and press Enter to display the Edit CICS Dataset Record screen (5.1.3).
15. Type **USING VSAM_EMP_RECORD** in the COMMAND field. **VSAM_EMP_RECORD** is the PLI 01 level that defines the DBUGEMP file in CWDEMPE (Figure 12-6).
16. Press Enter to map the data in this record to the data structure **VSAM_EMP_RECORD**.

    Notice the VALID COMMANDS field displays the commands that can be issued for this file. Any of these commands can be entered in the COMMAND field. You can browse through the file by using the NEXT command to move to the next record.

Figure 12-6. Edit CICS Dataset Record Screen (5.1.3)



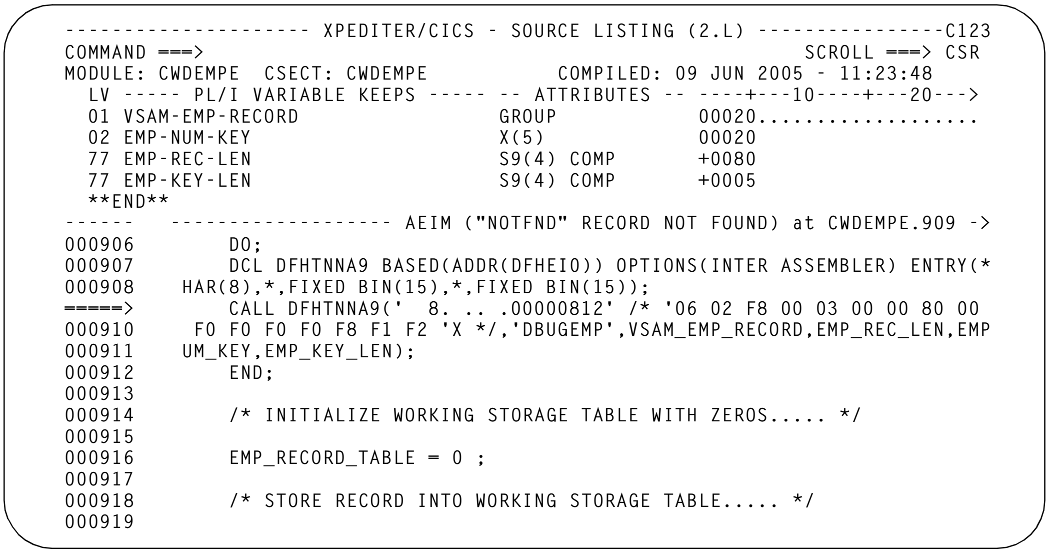
```
 -------------- XPEDITER/CICS - EDIT CICS DATASET RECORD (5.1.3) ----------C123
 COMMAND ===>                                                    SCROLL ===> CSR
 MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
 VALID COMMANDS: READ  NEXT  PREV  CLOSE
                        ACCESS METHOD: VSAM        TYPE: KSDS
 FILENAME: DBUGEMP                                RECLN: 00080  MAX RECLN: 00080
 DEC-OFFSET: 000000 ADD-OFFSET: ______   RECFM: F KEYLN: 00005        RKP: 00000
 KEY FIELD: ----5
            00010
            FFFFF
            00010

   FIELD LEVEL/NAME                  FORMAT      ----+---10----+---20----+---3>
   01 VSAM_EMP_RECORD                80/GRP
   02 EMP_NUM_KEY                    5/AN        00010
   02 EMP_NAME                       15/AN       Employee #10
   02 EMP_HOURS                      3/NUM       010
   02 EMP_TOTPAY                     7/NUM
   02 FILLER                         50/AN
   **END**
```

17. Type **NEXT** in the COMMAND field and press Enter to display the next record in the file.

    Look at EMP_NUM_KEY of the displayed record. Our example program abended because there were no records on the DBUGEMP file with the key 00002 (Figure 12-1 on page 12-2). However, there is a record with the key 00020.

18. Press PF6 (LOCATE *) to return to the Source Listing screen (2.L) to change the key.
19. Type **00020** in the data area of the field EMP_NUM_KEY as shown in Figure 12-7.
20. Press Enter to update the field.
21. Press PF12 (GO) to continue processing.

Figure 12-7. Modifying Key Data on the Source Listing Screen (2.L)

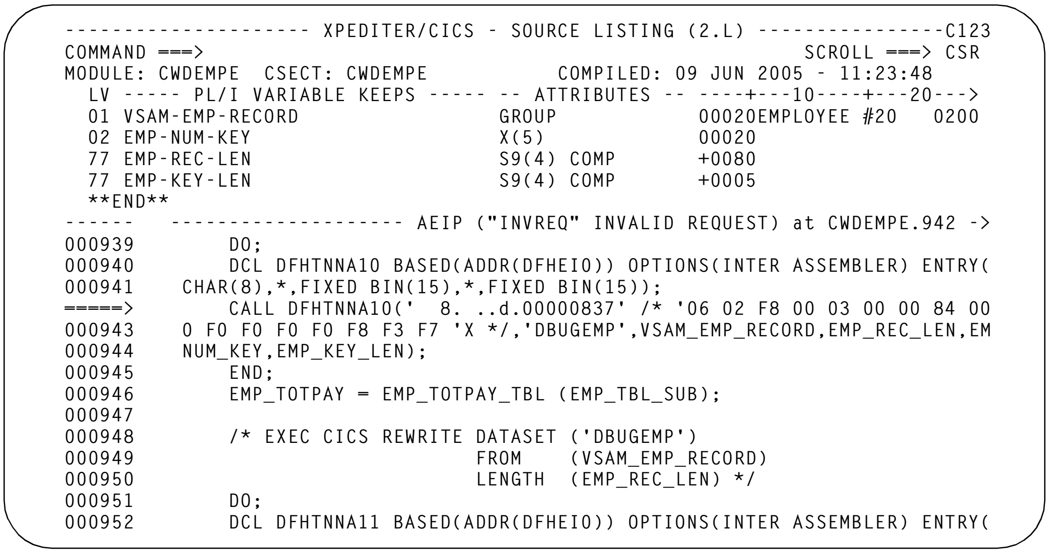


```
 -------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ---------------C123
 COMMAND ===>                                                    SCROLL ===> CSR
 MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
   LV ----- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
   01 VSAM-EMP-RECORD                 GROUP           00020..................
   02 EMP-NUM-KEY                     X(5)            00020
   77 EMP-REC-LEN                     S9(4) COMP      +0080
   77 EMP-KEY-LEN                     S9(4) COMP      +0005
   **END**
 ------   ------------------- AEIM ("NOTFND" RECORD NOT FOUND) at CWDEMPE.909 ->
 000906        DO;
 000907        DCL DFHTNNA9 BASED(ADDR(DFHEI0)) OPTIONS(INTER ASSEMBLER) ENTRY(*
 000908    HAR(8),*,FIXED BIN(15),*,FIXED BIN(15));
 =====>        CALL DFHTNNA9('  8. .. .00000812' /* '06 02 F8 00 03 00 00 80 00
 000910     F0 F0 F0 F0 F8 F1 F2 'X */,'DBUGEMP',VSAM_EMP_RECORD,EMP_REC_LEN,EMP
 000911    UM_KEY,EMP_KEY_LEN);
 000912        END;
 000913
 000914        /* INITIALIZE WORKING STORAGE TABLE WITH ZEROS..... */
 000915
 000916        EMP_RECORD_TABLE = 0 ;
 000917
 000918        /* STORE RECORD INTO WORKING STORAGE TABLE..... */
 000919
```

Changing File Service Requests

In this example, Xpediter/CICS intercepts an AEIP abend in the CWDEMPE program. An AEIP abend can be caused by many different problems. In this example, the last EXEC CICS command was a READ for UPDATE, as shown in lines 454 through 460 in Figure 12-8.

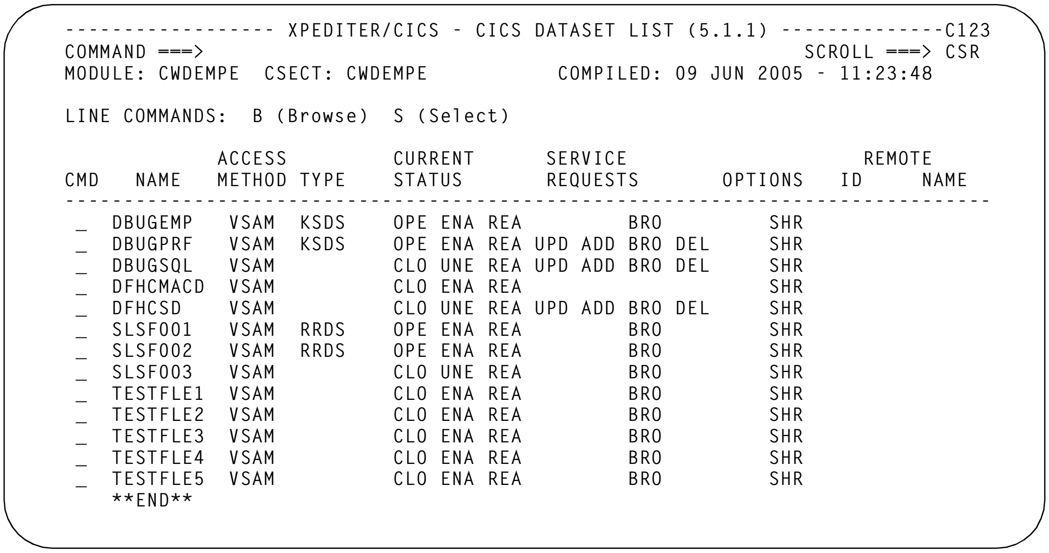
Figure 12-8. AEIP Abend on the Source Listing Screen (2.L)



```
-------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                     SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48
  LV ----- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
  01 VSAM-EMP-RECORD                  GROUP            00020EMPLOYEE #20   0200
  02 EMP-NUM-KEY                      X(5)             00020
  77 EMP-REC-LEN                      S9(4) COMP       +0080
  77 EMP-KEY-LEN                      S9(4) COMP       +0005
  **END**
------   -------------------- AEIP ("INVREQ" INVALID REQUEST) at CWDEMPE.942 ->
000939        DO;
000940        DCL DFHTNNA10 BASED(ADDR(DFHEI0)) OPTIONS(INTER ASSEMBLER) ENTRY(
000941    CHAR(8),*,FIXED BIN(15),*,FIXED BIN(15));
=====>        CALL DFHTNNA10('  8. ..d.00000837' /* '06 02 F8 00 03 00 00 84 00
000943    0 F0 F0 F0 F0 F8 F3 F7 'X */,'DBUGEMP',VSAM_EMP_RECORD,EMP_REC_LEN,EM
000944    NUM_KEY,EMP_KEY_LEN);
000945        END;
000946        EMP_TOTPAY = EMP_TOTPAY_TBL (EMP_TBL_SUB);
000947
000948        /* EXEC CICS REWRITE DATASET ('DBUGEMP')
000949                              FROM    (VSAM_EMP_RECORD)
000950                              LENGTH  (EMP_REC_LEN) */
000951        DO;
000952        DCL DFHTNNA11 BASED(ADDR(DFHEI0)) OPTIONS(INTER ASSEMBLER) ENTRY(
```

1. Type **=5.1.1** in the COMMAND field to check the service requests for DBUGEMP.
2. Press Enter. Xpediter/CICS transfers directly to the CICS Datasets List screen (5.1.1) (Figure 12-9).

Figure 12-9. CICS Dataset List Screen (5.1.1)



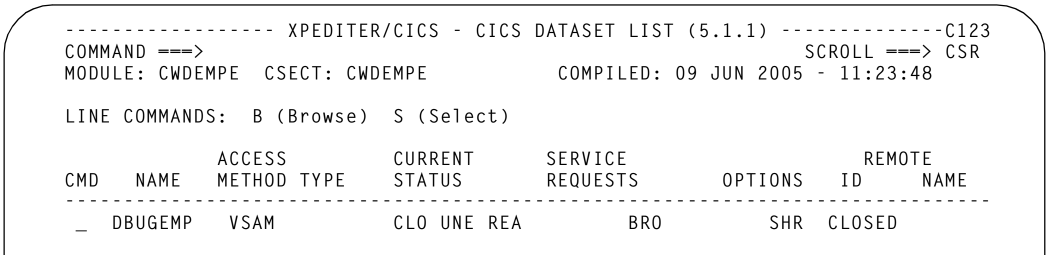
```
----------------- XPEDITER/CICS - CICS DATASET LIST (5.1.1) -------------C123
COMMAND ===>                                                     SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48

LINE COMMANDS:  B (Browse)  S (Select)

             ACCESS         CURRENT      SERVICE                   REMOTE
CMD   NAME   METHOD TYPE    STATUS       REQUESTS       OPTIONS   ID     NAME
-------------------------------------------------------------------------------
 _  DBUGEMP   VSAM  KSDS    OPE ENA REA         BRO         SHR
 _  DBUGPRF   VSAM  KSDS    OPE ENA REA UPD ADD BRO DEL     SHR
 _  DBUGSQL   VSAM          CLO UNE REA UPD ADD BRO DEL     SHR
 _  DFHCMACD  VSAM          CLO ENA REA                     SHR
 _  DFHCSD    VSAM          CLO UNE REA UPD ADD BRO DEL     SHR
 _  SLSF001   VSAM  RRDS    OPE ENA REA         BRO         SHR
 _  SLSF002   VSAM  RRDS    OPE ENA REA         BRO         SHR
 _  SLSF003   VSAM          CLO UNE REA         BRO         SHR
 _  TESTFLE1  VSAM          CLO ENA REA         BRO         SHR
 _  TESTFLE2  VSAM          CLO ENA REA         BRO         SHR
 _  TESTFLE3  VSAM          CLO ENA REA         BRO         SHR
 _  TESTFLE4  VSAM          CLO ENA REA         BRO         SHR
 _  TESTFLE5  VSAM          CLO ENA REA         BRO         SHR
    **END**
```

Look at the SERVICE REQUESTS field for DBUGEMP. Both read (REA) and browse (BRO) are specified for this file. There is no update (UPD) capability, so the read for update in CWDEMPE resulted in an INVALID REQUEST (AEIP) abend.

Authorized users can modify the current status and add or delete service requests (ADD, DELETE, BROWSE, UPDATE, or READ). To change the service request, you must first close and disable the dataset.

3. Type **CLO** in the CURRENT STATUS field for DBUGEMP.
4. Press Enter. As shown in Figure 12-10, the message CLOSED appears in the REMOTE field to indicate that the file has been closed. The CURRENT STATUS changes from OPE ENA to CLO UNE.

Figure 12-10. CLOSED Message

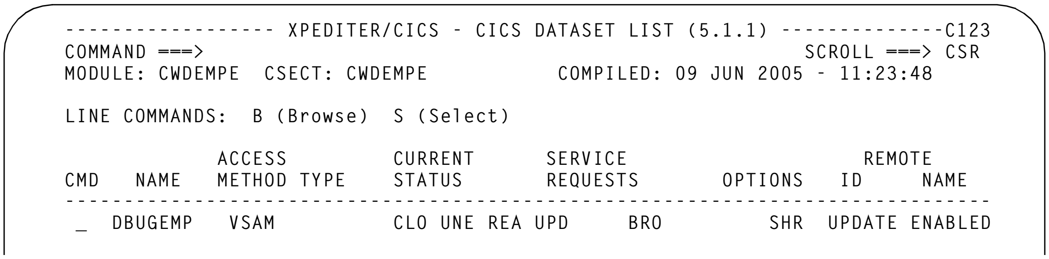


```
----------------- XPEDITER/CICS - CICS DATASET LIST (5.1.1) -------------C123
COMMAND ===>                                                SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48

LINE COMMANDS:  B (Browse)  S (Select)

             ACCESS         CURRENT      SERVICE                    REMOTE
CMD   NAME   METHOD TYPE    STATUS       REQUESTS       OPTIONS   ID     NAME
-------------------------------------------------------------------------------
 _  DBUGEMP   VSAM          CLO UNE REA         BRO         SHR  CLOSED
```

5. Type **UPD** in the SERVICE REQUESTS field next to REA.
6. Press Enter. Figure 12-11 shows that the message UPDATE ENABLED appears in the REMOTE field, indicating that update capabilities have been added.

   Note: This change is temporary and remains in effect until the region is recycled. You must update the file definition to make the change permanent.

Figure 12-11. UPDATE ENABLED Message

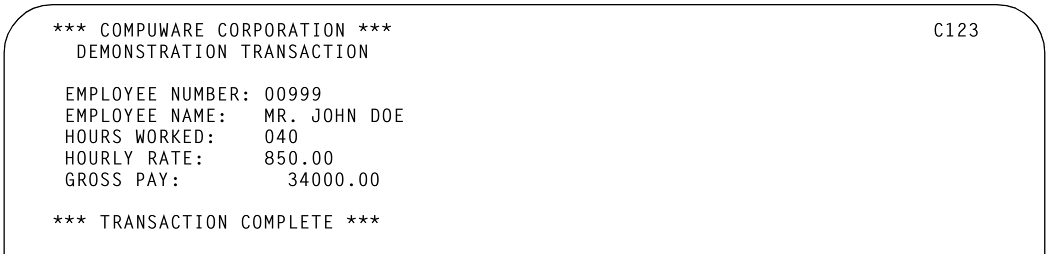


```
----------------- XPEDITER/CICS - CICS DATASET LIST (5.1.1) -------------C123
COMMAND ===>                                                SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48

LINE COMMANDS:  B (Browse)  S (Select)

             ACCESS         CURRENT      SERVICE                    REMOTE
CMD   NAME   METHOD TYPE    STATUS       REQUESTS       OPTIONS   ID     NAME
-------------------------------------------------------------------------------
 _  DBUGEMP   VSAM          CLO UNE REA UPD     BRO         SHR  UPDATE ENABLED
```

7. Type **OPE** in the CURRENT STATUS field to open the file.
8. Press Enter. The message OPEN is displayed in the REMOTE field to indicate that the file is open. The CURRENT STATUS changes from CLO UNE to OPE ENA.
9. Press PF12 (GO) to continue processing the transaction. The XPLE Transaction Complete screen appears as shown in Figure 12-12.

Remember to end your session as described in “Exiting Xpediter/CICS and Ending a Debugging Session” on page 2-6.

Figure 12-12. Transaction Complete Screen



```
*** COMPUWARE CORPORATION ***                                               C123
  DEMONSTRATION TRANSACTION

 EMPLOYEE NUMBER: 00999
 EMPLOYEE NAME:   MR. JOHN DOE
 HOURS WORKED:    040
 HOURLY RATE:     850.00
 GROSS PAY:         34000.00

*** TRANSACTION COMPLETE ***
```

Chapter 13.
Accessing DL/1 Databases

This chapter discusses how to use the Xpediter/CICS File Utility to access and modify IMS databases defined to your CICS region and/or the DBCTL region to which your CICS region is attached.

Note: A sample Compuware database was used to generate the screens shown in this chapter. Since the database you access will be different, your screens will vary from those shown. Use this chapter simply as a model of how to access your database.

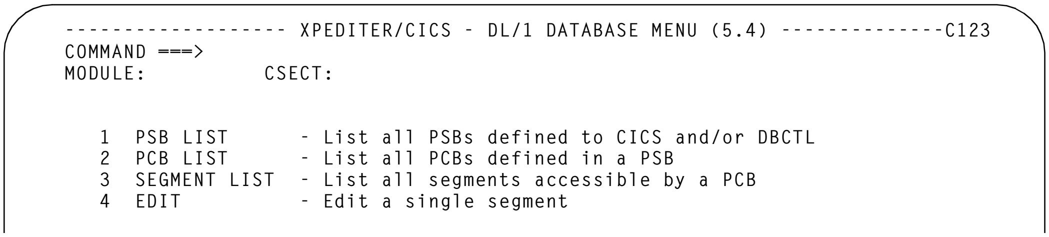
Note: The demonstrations in this chapter should be performed in Xpediter's standard operating mode — ***not*** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

Selecting PSBs and PCBs

1. From a blank CICS screen, type **XPED 5.4** and press Enter to display the DL/1 Database Menu (5.4) as shown in Figure 13-1.
2. Type **1** in the COMMAND field and press Enter to display the DL/1 PSB List screen (Figure 13-2), which lists the PSBs defined for use in this CICS region and the DBCTL region to which your CICS region is attached.

   Xpediter/CICS always presents a list of PSBs from which you can select, so you no longer have to supply complicated syntax to access the PSB. You just select the PSB that you want to work with.

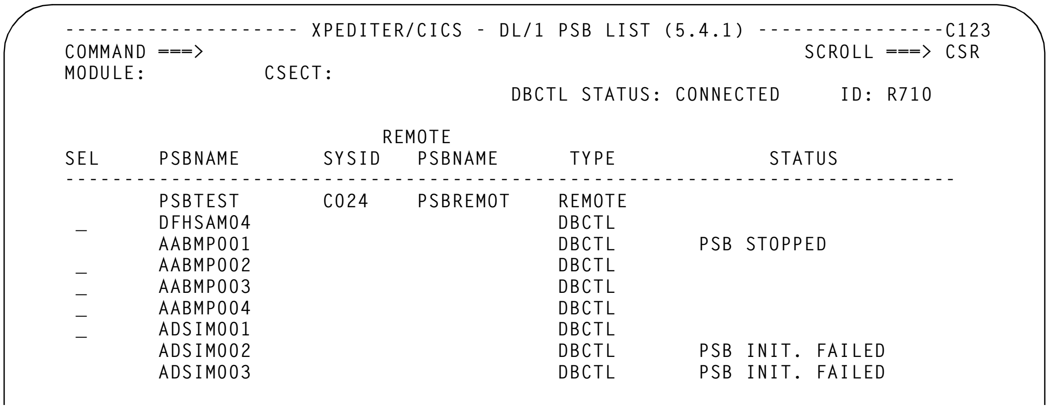
Figure 13-1. DL/1 Database Menu (5.4)



```
------------------ XPEDITER/CICS - DL/1 DATABASE MENU (5.4) -------------C123
COMMAND ===>
MODULE:          CSECT:


   1  PSB LIST      - List all PSBs defined to CICS and/or DBCTL
   2  PCB LIST      - List all PCBs defined in a PSB
   3  SEGMENT LIST  - List all segments accessible by a PCB
   4  EDIT          - Edit a single segment
```

Note that remote PSBs are shown first, listed with the remote system ID and the remote PSB name. Those remote PSBs are display-only and ***cannot*** be accessed from the File Utility. They are followed by DBCTL PSBs.

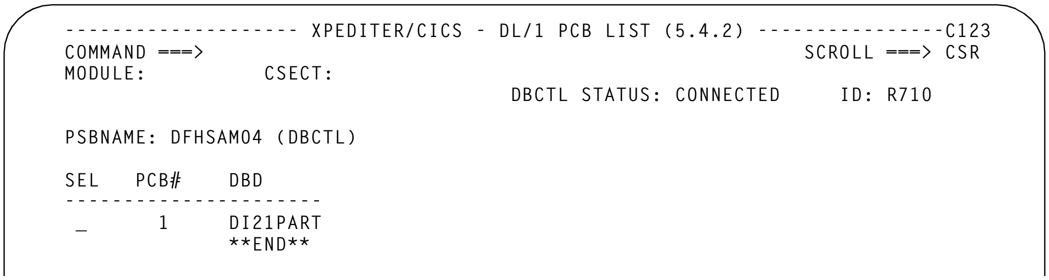
Figure 13-2. DL/1 PSB List Screen (5.4.1)



```
-------------------- XPEDITER/CICS - DL/1 PSB LIST (5.4.1) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
                                      DBCTL STATUS: CONNECTED     ID: R710

                           REMOTE
SEL     PSBNAME       SYSID   PSBNAME      TYPE             STATUS
----------------------------------------------------------------------------
        PSBTEST       C024    PSBREMOT    REMOTE
 _      DFHSAM04                          DBCTL
        AABMP001                          DBCTL       PSB STOPPED
 _      AABMP002                          DBCTL
 _      AABMP003                          DBCTL
 _      AABMP004                          DBCTL
 _      ADSIM001                          DBCTL
        ADSIM002                          DBCTL       PSB INIT. FAILED
        ADSIM003                          DBCTL       PSB INIT. FAILED
```

3. Type **S** in the SEL field next to the desired PSB name and press Enter to display a list of PCBs in the selected PSB. The DL/1 PCB List screen (5.4.2) (Figure 13-3) appears.

Figure 13-3. DL/1 PCB List Screen (5.4.2)



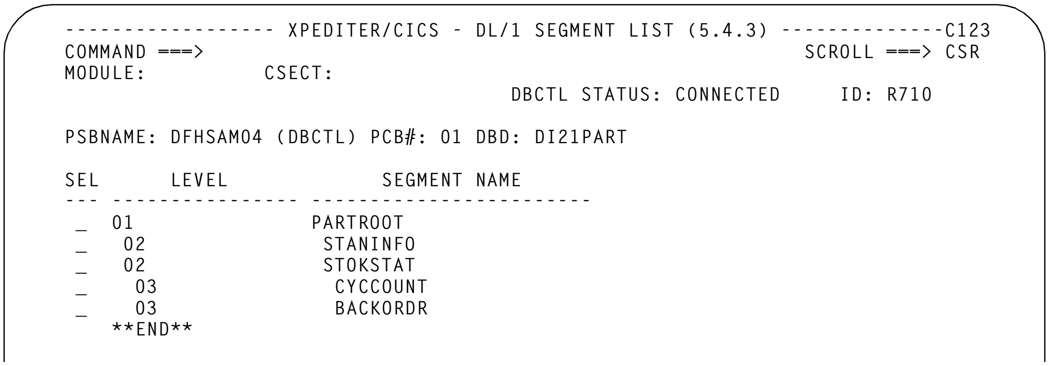
```
-------------------- XPEDITER/CICS - DL/1 PCB LIST (5.4.2) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
                                      DBCTL STATUS: CONNECTED     ID: R710

PSBNAME: DFHSAM04 (DBCTL)

SEL   PCB#    DBD
----------------------
 _      1     DI21PART
              **END**
```

A database description (DBD) is associated with a PCB and assigned a number which is displayed prior to the DBD name on this screen. This number is used in place of the DBD name, because multiple PCBs can access the same DBD using the same or slightly different views of the database. The PCB list is displayed in the order in which the PCBs have been defined in the PSB. You can scroll through this screen (UP, DOWN, TOP, BOTTOM), or you can position the cursor to a particular PCB using the LOCATE command followed by the DBD name.

Selecting Segments from the PCB

1. Type an **S** next to the selected PCB on the DL/1 PCB List screen, and press Enter to display the DL/1 Segment List screen (5.4.3) (Figure 13-4).

Figure 13-4. DL/1 Segment List Screen (5.4.3)



```
------------------ XPEDITER/CICS - DL/1 SEGMENT LIST (5.4.3) --------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
                                      DBCTL STATUS: CONNECTED     ID: R710

PSBNAME: DFHSAM04 (DBCTL) PCB#: 01 DBD: DI21PART

SEL      LEVEL             SEGMENT NAME
--- --------------- -----------------------
 _  01               PARTROOT
 _   02               STANINFO
 _   02               STOKSTAT
 _    03               CYCCOUNT
 _    03               BACKORDR
    **END**
```

This screen displays each segment that can be accessed via the selected PCB, along with the associated level number for that segment. Each level in the hierarchy is indented one position from the previous higher level to provide a hierarchical view of the database.

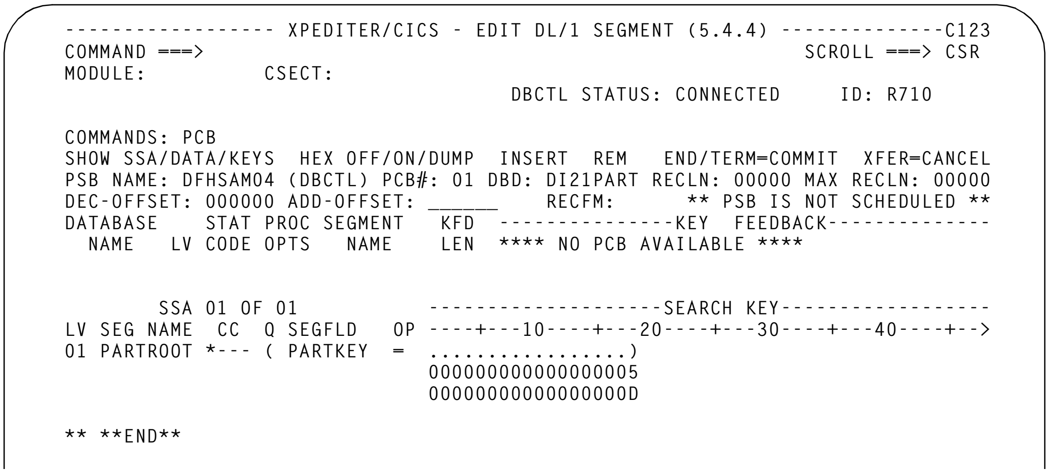
The P line command highlights the hierarchical path required to reach a specific segment.

2. Type a **P** command in the SEL field next to the lowest level segment to be accessed. Press Enter to highlight the path indicating how the segment must be accessed within the hierarchy.

   Note: The PSBNAME and PCB# fields on this screen can be used to directly access a segment list by typing the PSBNAME with a PCB number. Xpediter/CICS checks the PSB and PCB number and returns an error message when they are invalid.

3. Type an **S** in the SEL field next to a segment in this list and press Enter to display the Edit DL/1 Segment screen (5.4.4) shown in Figure 13-5.
4. Type **SHOW SSA** in the COMMAND field and press Enter. Xpediter/CICS creates a skeleton segment search argument (SSA).

Figure 13-5. Edit DL/1 Segment Screen (5.4.4)



```
------------------ XPEDITER/CICS - EDIT DL/1 SEGMENT (5.4.4) --------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
                                      DBCTL STATUS: CONNECTED     ID: R710

COMMANDS: PCB
SHOW SSA/DATA/KEYS  HEX OFF/ON/DUMP  INSERT  REM   END/TERM=COMMIT  XFER=CANCEL
PSB NAME: DFHSAM04 (DBCTL) PCB#: 01 DBD: DI21PART RECLN: 00000 MAX RECLN: 00000
DEC-OFFSET: 000000 ADD-OFFSET: ______    RECFM:      ** PSB IS NOT SCHEDULED **
DATABASE    STAT PROC SEGMENT   KFD  ---------------KEY  FEEDBACK--------------
  NAME   LV CODE OPTS   NAME    LEN  **** NO PCB AVAILABLE ****


        SSA 01 OF 01           --------------------SEARCH KEY------------------
LV SEG NAME  CC  Q SEGFLD   OP ----+---10----+---20----+---30----+---40----+-->
01 PARTROOT *--- ( PARTKEY  =  .................)
                               000000000000000005
                               00000000000000000D

** **END**
```

This screen is used to perform DL/1 calls to the database. Xpediter/CICS uses standard DL/1 notation rules to perform calls. READ and WRITE commands are not used. Xpediter/CICS can build skeleton SSAs to access a segment or rebuild complete

SSAs displaying the path to the segment. You can perform sequential and random calls to a database using PCB and TERM calls. A PSB holds the position in the database for up to two minutes. That time is the default value of the global parameter PSBWAIT and can be changed by specifying another value between 1 and 59.

The screen also shows the valid commands that are specified in the PCB definition. The commands indicate the functions that can be used with DL/1 segments, I/O area manipulation, and screen display. If NONE shows in the VALID COMMANDS area, either Xpediter/CICS cannot determine the valid DL/1 commands or you are not authorized to perform functions on this screen.

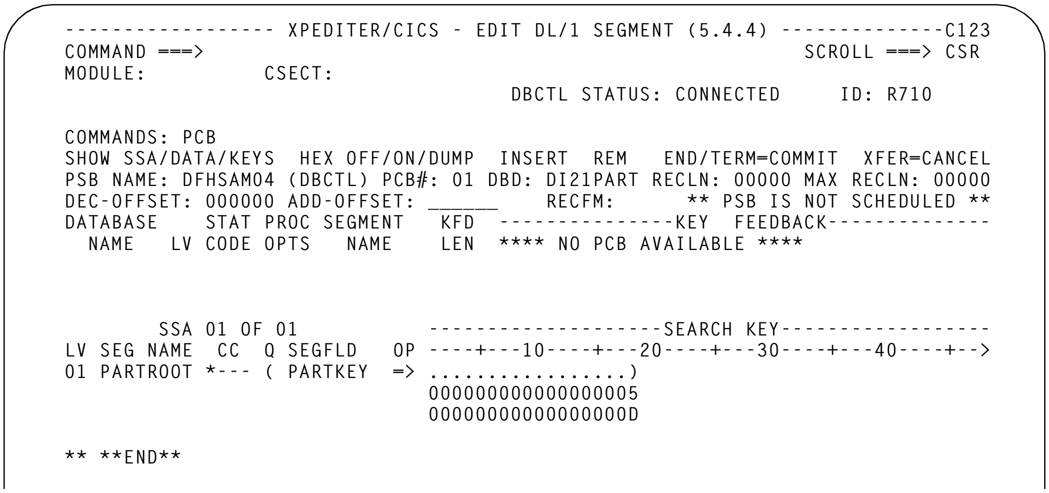
In this example, no PSB has been scheduled. The next example shows how to schedule a PSB and retrieve a DL/1 segment.

Retrieving a DL/1 Segment

A qualification statement provides DL/1 with information about a specific segment occurrence. You provide DL/1 with the name of a field in the segment and a value for the specific field. The field and value are connected by a relational operator (OP) that tells DL/1 how to compare the two values.

1. Type a greater-than symbol (>) in the OP field to the right of the equal sign (=). Press Enter to update the OP field. This tells Xpediter/CICS to search for a segment with a value in PARTKEY greater than or equal to low values, such as the first segment in the database (Figure 13-6).

Figure 13-6. Modifying the SSA on the Edit DL/1 Segment Screen (5.4.4)



```
------------------ XPEDITER/CICS - EDIT DL/1 SEGMENT (5.4.4) --------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE:          CSECT:
                                      DBCTL STATUS: CONNECTED     ID: R710

COMMANDS: PCB
SHOW SSA/DATA/KEYS  HEX OFF/ON/DUMP  INSERT  REM   END/TERM=COMMIT  XFER=CANCEL
PSB NAME: DFHSAM04 (DBCTL) PCB#: 01 DBD: DI21PART RECLN: 00000 MAX RECLN: 00000
DEC-OFFSET: 000000 ADD-OFFSET: ______     RECFM:      ** PSB IS NOT SCHEDULED **
DATABASE    STAT PROC SEGMENT   KFD  ---------------KEY  FEEDBACK--------------
  NAME   LV CODE OPTS   NAME    LEN  **** NO PCB AVAILABLE ****



        SSA 01 OF 01           --------------------SEARCH KEY------------------
LV SEG NAME  CC  Q SEGFLD   OP ----+---10----+---20----+---30----+---40----+-->
01 PARTROOT *--- ( PARTKEY  => ..................)
                               000000000000000005
                               00000000000000000D

** **END**
```

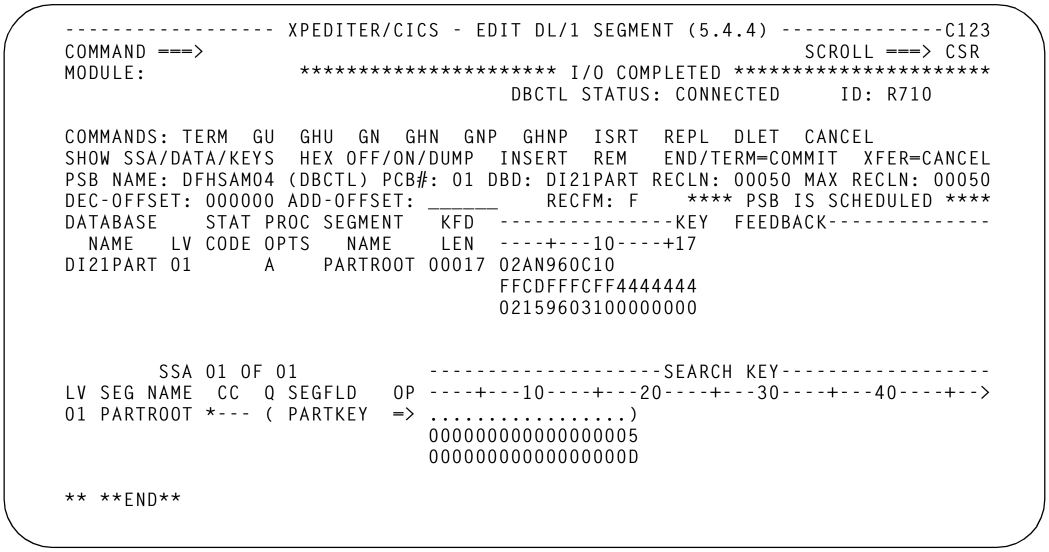
2. Type **PCB** and press Enter to schedule a PSB. Xpediter/CICS displays the message

```
************* PSB SCHEDULED VIA "PCB" COMMAND *************
```

   to show that the PSB has been scheduled.

3. Type **GN** in the COMMAND field and press Enter to display the next segment in the database. The Edit DL/1 Segment screen (5.4.4) appears with the PCB field area updated for the selected segment (Figure 13-7).

Figure 13-7. Displaying the Area on the PCB Edit DL/1 Segment Screen (5.4.4)



```
------------------ XPEDITER/CICS - EDIT DL/1 SEGMENT (5.4.4) --------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:             ********************** I/O COMPLETED **********************
                                      DBCTL STATUS: CONNECTED     ID: R710

COMMANDS: TERM  GU  GHU  GN  GHN  GNP  GHNP  ISRT  REPL  DLET  CANCEL
SHOW SSA/DATA/KEYS  HEX OFF/ON/DUMP  INSERT  REM   END/TERM=COMMIT  XFER=CANCEL
PSB NAME: DFHSAM04 (DBCTL) PCB#: 01 DBD: DI21PART RECLN: 00050 MAX RECLN: 00050
DEC-OFFSET: 000000 ADD-OFFSET: ______    RECFM: F    **** PSB IS SCHEDULED ****
DATABASE    STAT PROC SEGMENT   KFD  ---------------KEY  FEEDBACK--------------
  NAME   LV CODE OPTS   NAME    LEN  ----+---10----+17
DI21PART 01      A    PARTROOT 00017 02AN960C10
                                     FFCDFFFCFF4444444
                                     02159603100000000


   SSA 01 OF 01                --------------------SEARCH KEY------------------
LV SEG NAME  CC  Q SEGFLD   OP ----+---10----+---20----+---30----+---40----+-->
01 PARTROOT *--- ( PARTKEY  => ..................)
                               000000000000000005
                               00000000000000000D

** **END**
```

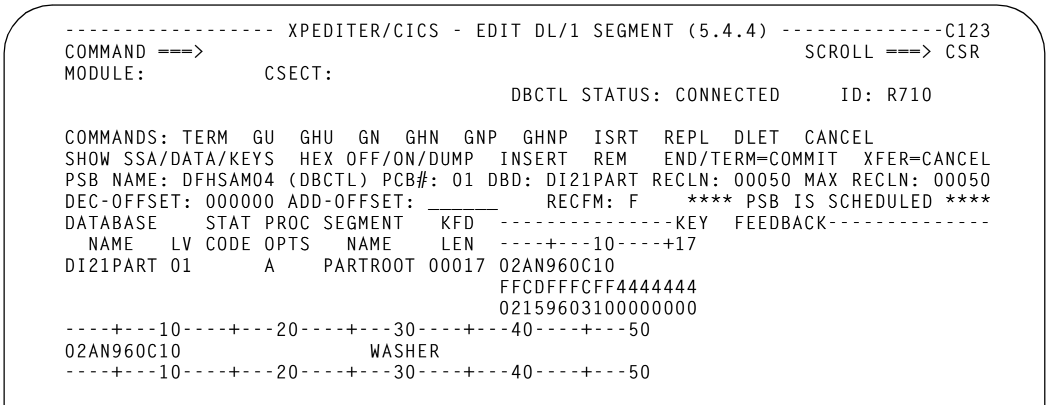
The PCB field area displays data fields obtained from the PCB used in the last DL/1 call. These fields reflect the current position in the database and the status returned by DL/1.

The DBD field identifies DI21PART as the database being accessed. The LV and SEGMENT NAME fields indicate the lowest segment in the last path DL/1 encountered while searching for the requested segment. The blanks in the STAT CODE field indicate that the call was successful. If there was an error in processing, this field would display a two-character status code, such as GB, AK or NO. In addition, Xpediter/CICS provides extended diagnostics for many of the displayed status codes. You can view these diagnostics by typing **HELP *xx*** in the COMMAND field, where *xx* is the DL/1 status code.

The value in the PROC OPTS field indicates the type of call that can be issued by this PCB. The A value indicates that all types of calls can be issued. A **G** value would indicate “get processing” calls.

4. To display the data retrieved in this call, type **SHOW DATA** on the COMMAND field and press Enter. The SSA area at the bottom of the screen is replaced by the segment data (Figure 13-8).

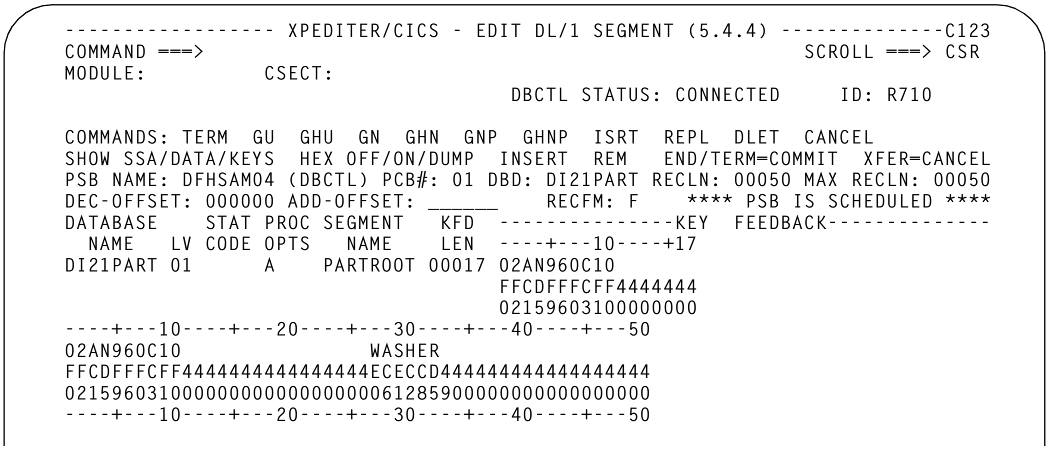
Figure 13-8. Edit DL/1 Segment Screen (5.4.4) - SHOW DATA



```
------------------ XPEDITER/CICS - EDIT DL/1 SEGMENT (5.4.4) --------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
                                      DBCTL STATUS: CONNECTED     ID: R710

COMMANDS: TERM  GU  GHU  GN  GHN  GNP  GHNP  ISRT  REPL  DLET  CANCEL
SHOW SSA/DATA/KEYS  HEX OFF/ON/DUMP  INSERT  REM   END/TERM=COMMIT  XFER=CANCEL
PSB NAME: DFHSAM04 (DBCTL) PCB#: 01 DBD: DI21PART RECLN: 00050 MAX RECLN: 00050
DEC-OFFSET: 000000 ADD-OFFSET: ______    RECFM: F    **** PSB IS SCHEDULED ****
DATABASE    STAT PROC SEGMENT   KFD  ---------------KEY  FEEDBACK--------------
  NAME   LV CODE OPTS   NAME    LEN  ----+---10----+17
DI21PART 01      A    PARTROOT 00017 02AN960C10
                                     FFCDFFFCFF4444444
                                     02159603100000000
----+---10----+---20----+---30----+---40----+---50
02AN960C10                WASHER
----+---10----+---20----+---30----+---40----+---50
```

5. The segment information can be displayed in several different formats to make editing easier. Use the HEX ON and HEX DUMP commands to format the display as shown in Figure 13-9 and Figure 13-10. HEX OFF returns the display to character format.

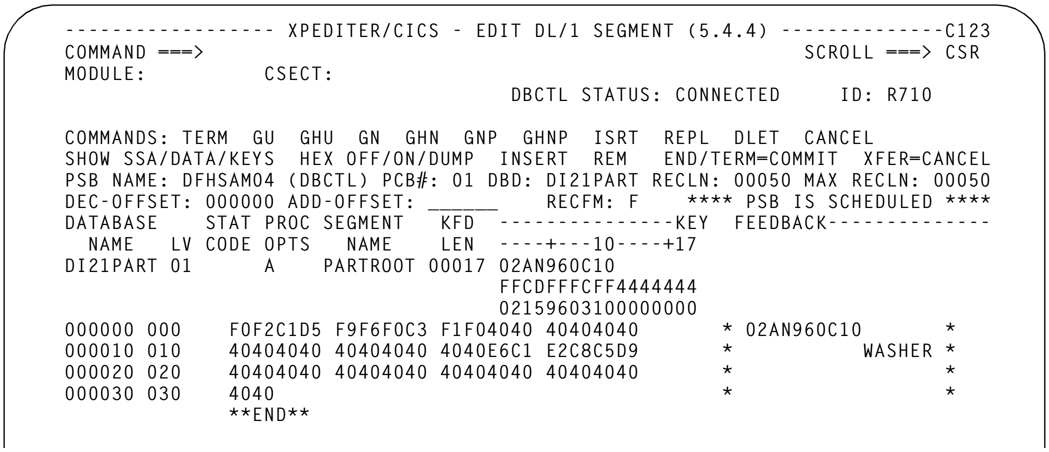
Figure 13-9. Edit DL/1 Segment Screen (5.4.4) - HEX ON



```
----------------- XPEDITER/CICS - EDIT DL/1 SEGMENT (5.4.4) -------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
                                      DBCTL STATUS: CONNECTED     ID: R710

COMMANDS: TERM  GU  GHU  GN  GHN  GNP  GHNP  ISRT  REPL  DLET  CANCEL
SHOW SSA/DATA/KEYS  HEX OFF/ON/DUMP  INSERT  REM   END/TERM=COMMIT  XFER=CANCEL
PSB NAME: DFHSAM04 (DBCTL) PCB#: 01 DBD: DI21PART RECLN: 00050 MAX RECLN: 00050
DEC-OFFSET: 000000 ADD-OFFSET: ______    RECFM: F    **** PSB IS SCHEDULED ****
DATABASE    STAT PROC SEGMENT   KFD  ---------------KEY  FEEDBACK-------------
  NAME   LV CODE OPTS   NAME    LEN  ----+---10----+17
DI21PART 01      A    PARTROOT 00017 02AN960C10
                                     FFCDFFFCFF4444444
                                     02159603100000000
----+---10----+---20----+---30----+---40----+---50
02AN960C10                WASHER
FFCDFFFCFF4444444444444444ECECCD444444444444444444
02159603100000000000000000612859000000000000000000
----+---10----+---20----+---30----+---40----+---50
```

6. To redisplay the original SSA, type **SHOW SSA** on the COMMAND line and press Enter.

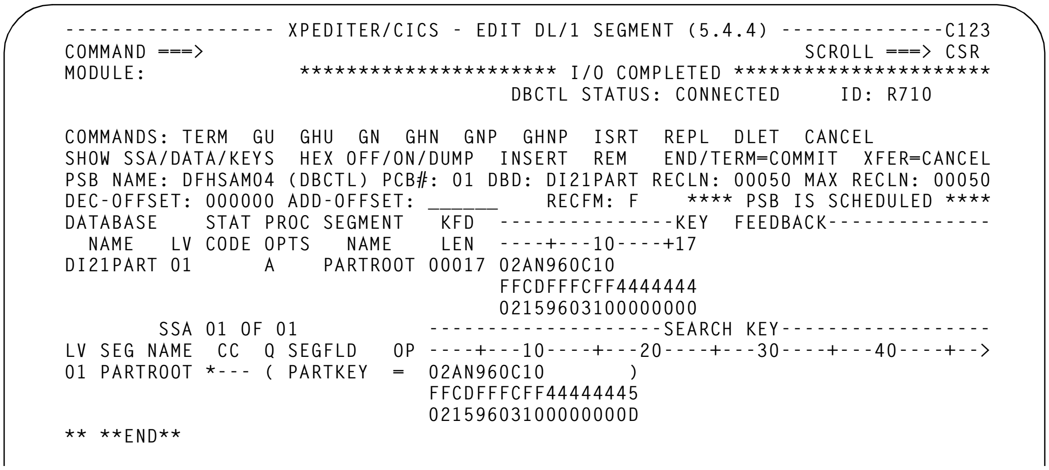
Figure 13-10. Edit DL/1 Segment Screen (5.4.4) - HEX DUMP



```
----------------- XPEDITER/CICS - EDIT DL/1 SEGMENT (5.4.4) -------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
                                      DBCTL STATUS: CONNECTED     ID: R710

COMMANDS: TERM  GU  GHU  GN  GHN  GNP  GHNP  ISRT  REPL  DLET  CANCEL
SHOW SSA/DATA/KEYS  HEX OFF/ON/DUMP  INSERT  REM   END/TERM=COMMIT  XFER=CANCEL
PSB NAME: DFHSAM04 (DBCTL) PCB#: 01 DBD: DI21PART RECLN: 00050 MAX RECLN: 00050
DEC-OFFSET: 000000 ADD-OFFSET: ______    RECFM: F    **** PSB IS SCHEDULED ****
DATABASE    STAT PROC SEGMENT   KFD  ---------------KEY  FEEDBACK-------------
  NAME   LV CODE OPTS   NAME    LEN  ----+---10----+17
DI21PART 01      A    PARTROOT 00017 02AN960C10
                                     FFCDFFFCFF4444444
                                     02159603100000000
000000 000    F0F2C1D5 F9F6F0C3 F1F04040 40404040      * 02AN960C10       *
000010 010    40404040 40404040 4040E6C1 E2C8C5D9      *           WASHER *
000020 020    40404040 40404040 40404040 40404040      *                  *
000030 030    4040                                     *                  *
              **END**
```

7. To update the SSA with key feedback data from the segment, type **SHOW KEYS** on the COMMAND line and press Enter. This is used to update the SSA to the current segment (Figure 13-11).

Figure 13-11. Edit DL/1 Segment Screen (5.4.4) - SHOW KEYS



```
------------------ XPEDITER/CICS - EDIT DL/1 SEGMENT (5.4.4) --------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:             ********************** I/O COMPLETED **********************
                                      DBCTL STATUS: CONNECTED     ID: R710

COMMANDS: TERM  GU  GHU  GN  GHN  GNP  GHNP  ISRT  REPL  DLET  CANCEL
SHOW SSA/DATA/KEYS  HEX OFF/ON/DUMP  INSERT  REM   END/TERM=COMMIT  XFER=CANCEL
PSB NAME: DFHSAM04 (DBCTL) PCB#: 01 DBD: DI21PART RECLN: 00050 MAX RECLN: 00050
DEC-OFFSET: 000000 ADD-OFFSET: ______   RECFM: F    **** PSB IS SCHEDULED ****
DATABASE    STAT PROC SEGMENT   KFD  ---------------KEY  FEEDBACK--------------
  NAME   LV CODE OPTS   NAME    LEN  ----+---10----+17
DI21PART 01      A    PARTROOT 00017 02AN960C10
                                     FFCDFFFCFF4444444
                                     02159603100000000
        SSA 01 OF 01           --------------------SEARCH KEY------------------
LV SEG NAME  CC  Q SEGFLD   OP ----+---10----+---20----+---30----+---40----+-->
01 PARTROOT *--- ( PARTKEY  =  02AN960C10       )
                               FFCDFFFCFF44444445
                               0215960310000000D
** **END**
```

8. The GN (GETNEXT) command can be used to browse the database.

9. Remember to end the session as described in “Exiting Xpediter/CICS and Ending a Debugging Session” on page 2-6.

Chapter 14. Using Xpediter/CICS with DB2

Xpediter/CICS provides support for IBM's DB2 relational database manager. In addition to the extensive interactive debugging facilities available to all CICS programs, special facilities have been created to meet the needs of the DB2 programmer.

Note: The demonstrations in this chapter should be performed in Xpediter's standard operating mode — ***not*** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

Using the DB2 Easy Query

Xpediter/CICS provides selection lists that specify the columns to be used, then generates SQL calls and passes them to DB2 to execute. The DB2 File Utility in Xpediter/CICS honors all DB2 security and referential integrity rules. You can use the DB2 file utility to update only the tables you have authority to access.

Note: A sample Compuware database was used to generate the screens shown in this chapter. Since the database you access will be different, your screens will vary from those shown. Use this chapter simply as a model of how to access your database.

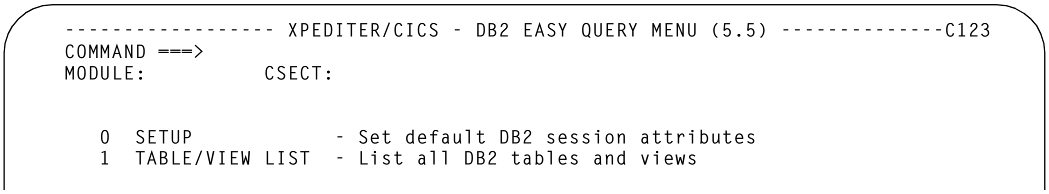
In order to limit resource contention, access to the DB2 File Utility is prohibited from a terminal that has a program in a BREAK/ABEND state.

Setting DB2 Session Default Attributes

The DB2 Setup screen lets you override default DB2 parameters established during installation.

1. From a blank CICS screen, type **XPED 5.5** and press Enter to access the DB2 Easy Query Menu (Figure 14-1).

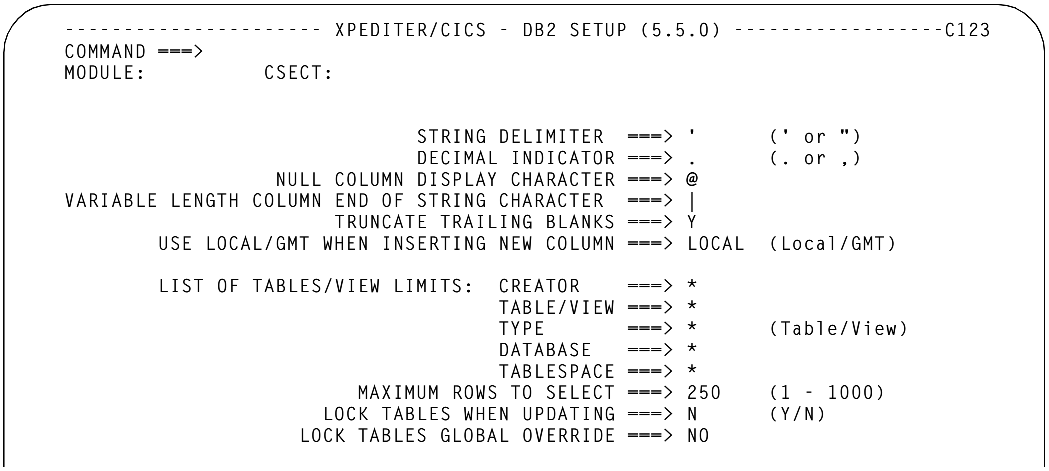
Figure 14-1. DB2 Easy Query Menu (5.5)



```
----------------- XPEDITER/CICS - DB2 EASY QUERY MENU (5.5) -------------C123
COMMAND ===>
MODULE:          CSECT:


   0  SETUP             - Set default DB2 session attributes
   1  TABLE/VIEW LIST  - List all DB2 tables and views
```

2. Type **0** and press Enter to display the DB2 Setup screen (5.5.0) (Figure 14-2).

Figure 14-2. DB2 Setup Screen (5.5.0)



```
---------------------- XPEDITER/CICS - DB2 SETUP (5.5.0) ------------------C123
COMMAND ===>
MODULE:          CSECT:


                              STRING DELIMITER  ===> '      (' or ")
                              DECIMAL INDICATOR ===> .      (. or ,)
                  NULL COLUMN DISPLAY CHARACTER ===> @
VARIABLE LENGTH COLUMN END OF STRING CHARACTER  ===> |
                       TRUNCATE TRAILING BLANKS ===> Y
        USE LOCAL/GMT WHEN INSERTING NEW COLUMN ===> LOCAL  (Local/GMT)

        LIST OF TABLES/VIEW LIMITS:  CREATOR    ===> *
                                     TABLE/VIEW ===> *
                                     TYPE       ===> *      (Table/View)
                                     DATABASE   ===> *
                                     TABLESPACE ===> *
                         MAXIMUM ROWS TO SELECT ===> 250    (1 - 1000)
                      LOCK TABLES WHEN UPDATING ===> N      (Y/N)
                    LOCK TABLES GLOBAL OVERRIDE ===> NO
```

It is unlikely that you will change the first six fields shown on this screen. The LIST OF TABLES/VIEW LIMITS fields are used to limit the tables displayed on the DB2 Table/View List screen (5.5.1). The MAXIMUM ROWS TO SELECT field limits the number of rows retrieved during a browse or query. This value is set during installation and may be changed. The maximum value, which defaults to 1,000, is set at product installation time and can't be exceeded.

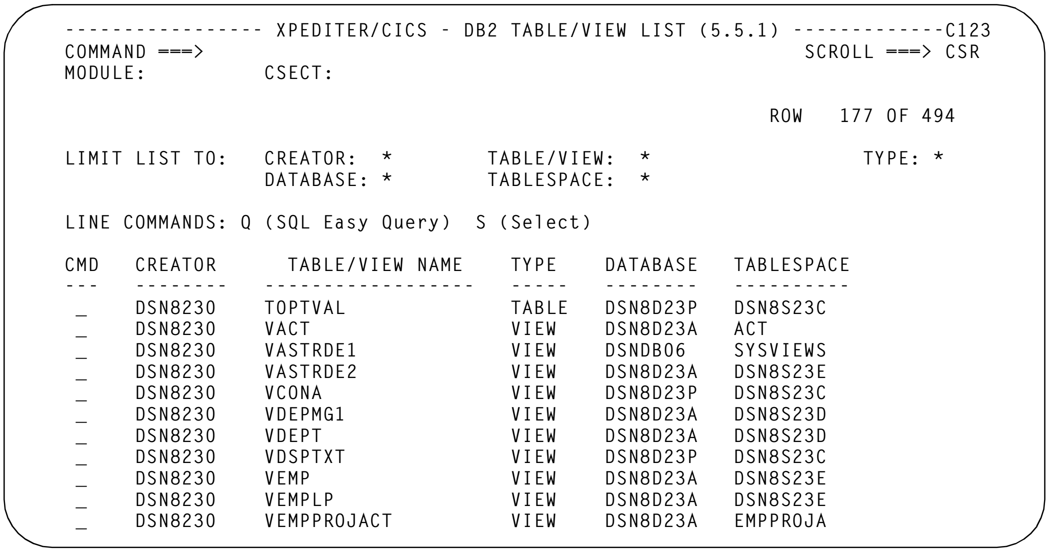
The LOCK TABLES WHEN UPDATING field places or prohibits a lock on a table selected for update. Specifying Y means that others cannot modify the table while you access it. If you specify N in this field, you risk losing changes, but you gain resource savings. For this reason, the system programmer can disable the lock capability. If the lock capability is disabled, the LOCK TABLES GLOBAL OVERRIDE field is set to NO.

3. To change a value on this screen, type over the existing value and press Enter. If the Xpediter/CICS profile dataset is used, the overrides are saved for future debugging sessions.

Accessing a List of DB2 Tables

1. Type **=5.5.1** in the COMMAND field of any Xpediter/CICS screen and press Enter to transfer to the DB2 Table/View List screen (5.5.1) (Figure 14-3).

Figure 14-3. DB2 Table/View List Screen (5.5.1)



```
---------------- XPEDITER/CICS - DB2 TABLE/VIEW LIST (5.5.1) -------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:

                                                            ROW   177 OF 494

LIMIT LIST TO:   CREATOR:  *        TABLE/VIEW:  *                  TYPE: *
                 DATABASE: *        TABLESPACE:  *

LINE COMMANDS: Q (SQL Easy Query)  S (Select)

CMD   CREATOR      TABLE/VIEW NAME    TYPE    DATABASE   TABLESPACE
---   --------   ------------------   -----   --------   ----------
 _    DSN8230    TOPTVAL              TABLE   DSN8D23P   DSN8S23C
 _    DSN8230    VACT                 VIEW    DSN8D23A   ACT
 _    DSN8230    VASTRDE1             VIEW    DSNDB06    SYSVIEWS
 _    DSN8230    VASTRDE2             VIEW    DSN8D23A   DSN8S23E
 _    DSN8230    VCONA                VIEW    DSN8D23P   DSN8S23C
 _    DSN8230    VDEPMG1              VIEW    DSN8D23A   DSN8S23D
 _    DSN8230    VDEPT                VIEW    DSN8D23A   DSN8S23D
 _    DSN8230    VDSPTXT              VIEW    DSN8D23P   DSN8S23C
 _    DSN8230    VEMP                 VIEW    DSN8D23A   DSN8S23E
 _    DSN8230    VEMPLP               VIEW    DSN8D23A   DSN8S23E
 _    DSN8230    VEMPPROJACT          VIEW    DSN8D23A   EMPPROJA
```

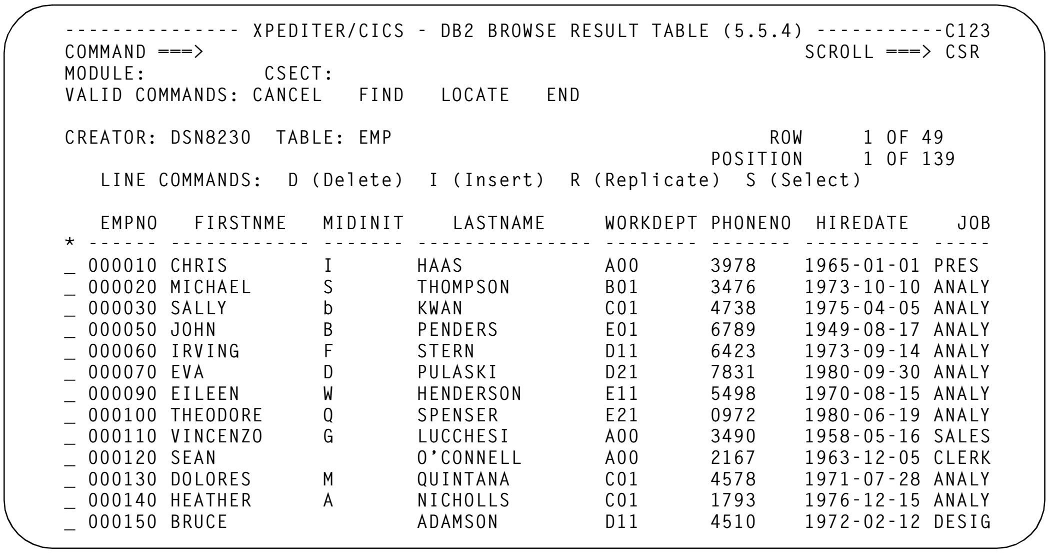
The DB2 Table/View List screen (5.5.1) displays a list of DB2 tables and views you are authorized to access. The list is in alphabetical order by creator. DB2 security limits the list to tables and views you are allowed to access with your CICS user ID.

The list can be further restricted by entering CREATOR, DATABASE, TABLE/VIEW, TABLESPACE, or TYPE in the LIMIT LIST TO fields. If these fields were specified on the DB2 Setup screen (5.5.0), these values are carried forward and appear on this screen.

There are two choices from this screen: Q to create an SQL Easy Query or S to select a table or view on which to work.

2. Type the **S** line command in the CMD field next to any table and press Enter to display the DB2 Browse Result Table screen (5.5.4) (Figure 14-4).

Figure 14-4. DB2 Browse Result Table Screen (5.5.4)



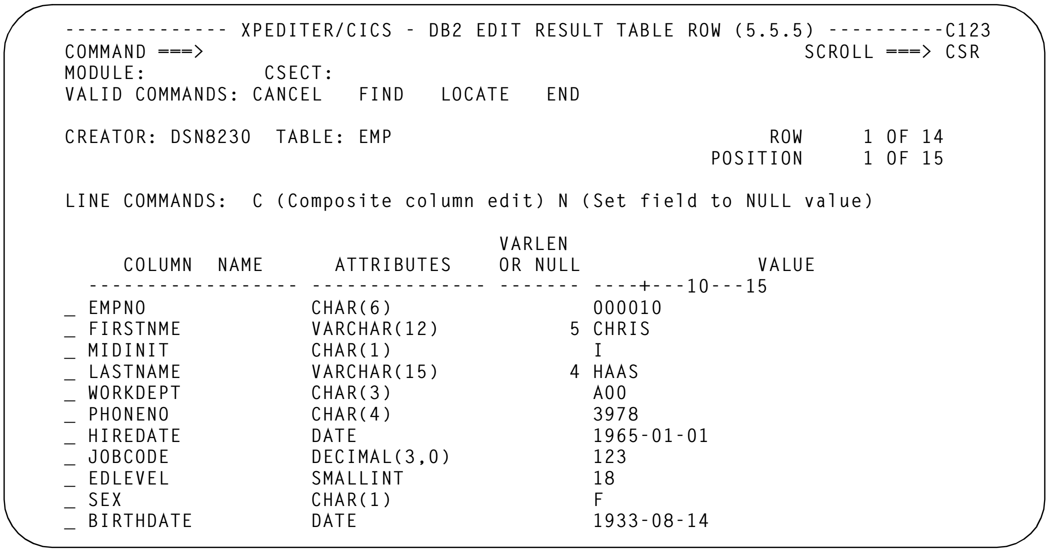
```
--------------- XPEDITER/CICS - DB2 BROWSE RESULT TABLE (5.5.4) -----------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
VALID COMMANDS: CANCEL   FIND   LOCATE   END

CREATOR: DSN8230  TABLE: EMP                                ROW     1 OF 49
                                                       POSITION     1 OF 139
   LINE COMMANDS:  D (Delete)  I (Insert)  R (Replicate)  S (Select)

   EMPNO   FIRSTNME   MIDINIT    LASTNAME     WORKDEPT PHONENO  HIREDATE    JOB
* ------ ------------ ------- --------------- -------- ------- ---------- -----
_ 000010 CHRIS        I       HAAS            A00      3978    1965-01-01 PRES
_ 000020 MICHAEL      S       THOMPSON        B01      3476    1973-10-10 ANALY
_ 000030 SALLY        b       KWAN            C01      4738    1975-04-05 ANALY
_ 000050 JOHN         B       PENDERS         E01      6789    1949-08-17 ANALY
_ 000060 IRVING       F       STERN           D11      6423    1973-09-14 ANALY
_ 000070 EVA          D       PULASKI         D21      7831    1980-09-30 ANALY
_ 000090 EILEEN       W       HENDERSON       E11      5498    1970-08-15 ANALY
_ 000100 THEODORE     Q       SPENSER         E21      0972    1980-06-19 ANALY
_ 000110 VINCENZO     G       LUCCHESI        A00      3490    1958-05-16 SALES
_ 000120 SEAN                 O’CONNELL       A00      2167    1963-12-05 CLERK
_ 000130 DOLORES      M       QUINTANA        C01      4578    1971-07-28 ANALY
_ 000140 HEATHER      A       NICHOLLS        C01      1793    1976-12-15 ANALY
_ 000150 BRUCE                ADAMSON         D11      4510    1972-02-12 DESIG
```

This screen displays the rows in the selected table. The display is by column name, and the rows are automatically formatted. The ROW field shows the current top row and the number of rows in the resulting table. The total rows may be limited by the MAXIMUM ROWS TO SELECT parameter on the DB2 Setup screen (5.5.0). The FIND and LOCATE commands can be used to find a string or shift the display to a particular column.

Editing a Row

1. Type **S** next to any row displayed on the DB2 Browse Result Table screen (5.5.4) and press Enter. The DB2 Edit Result Table Row screen (5.5.5) appears as shown in Figure 14-5.

Figure 14-5. DB2 Edit Result Table Row Screen (5.5.5)



```
------------- XPEDITER/CICS - DB2 EDIT RESULT TABLE ROW (5.5.5) ----------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
VALID COMMANDS: CANCEL   FIND   LOCATE   END

CREATOR: DSN8230  TABLE: EMP                                ROW     1 OF 14
                                                       POSITION     1 OF 15

LINE COMMANDS:  C (Composite column edit) N (Set field to NULL value)

                                     VARLEN
     COLUMN  NAME      ATTRIBUTES    OR NULL               VALUE
  ----------------- -------------- ------- ----+---10---15
_ EMPNO               CHAR(6)                000010
_ FIRSTNME            VARCHAR(12)          5 CHRIS
_ MIDINIT             CHAR(1)                I
_ LASTNAME            VARCHAR(15)          4 HAAS
_ WORKDEPT            CHAR(3)                A00
_ PHONENO             CHAR(4)                3978
_ HIREDATE            DATE                   1965-01-01
_ JOBCODE             DECIMAL(3,0)           123
_ EDLEVEL             SMALLINT               18
_ SEX                 CHAR(1)                F
_ BIRTHDATE           DATE                   1933-08-14
```

This screen is used to edit data in the selected row of the DB2 result table. Data in all columns may be updated. If you are not authorized to update a column by DB2 security, the contents of the column are protected to prevent modification.

2. You can move to a specific column by using the FIND command to position to a data string, or LOCATE to position to a column name. Two line commands are also available: N sets a field to null, and C transfers to the DB2 Edit Composite Column screen (5.5.6) described in "Editing a Column" on page 14-9.

   The DB2 File Utility has built-in edit functions to verify data.

3. Position the cursor to the VALUE field in a column with a DECIMAL attribute. Overtype the displayed value, and type an additional character. In this sample, we attempted to enter a four-character value in the JOBCODE field defined as DECIMAL (3,0).

4. Press Enter. Xpediter/CICS displays an error message

```
+++++++++++ INTEGER TO DECIMAL CONVERSION ERROR +++++++++++
```

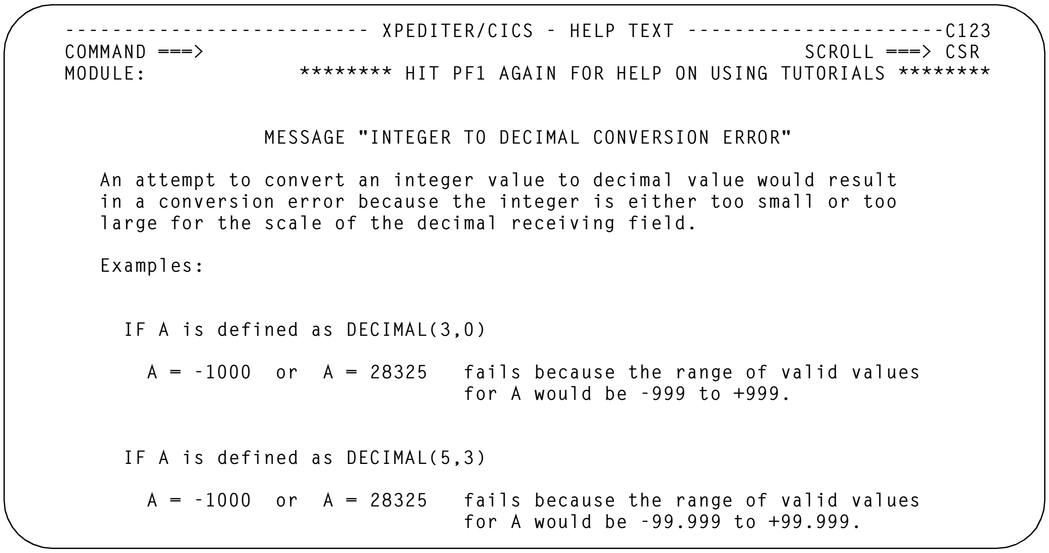
   to show that the data was entered incorrectly.

   Xpediter/CICS provides specific diagnostics for many DB2 errors.

   Note: The plus signs (+++) preceding and following the message indicate that additional detail information is available using the Help facility.

5. Press PF1 (HELP) to access the Help screen (Figure 14-6).

Figure 14-6. Help Screen for INTEGER TO DECIMAL CONVERSION ERROR Message



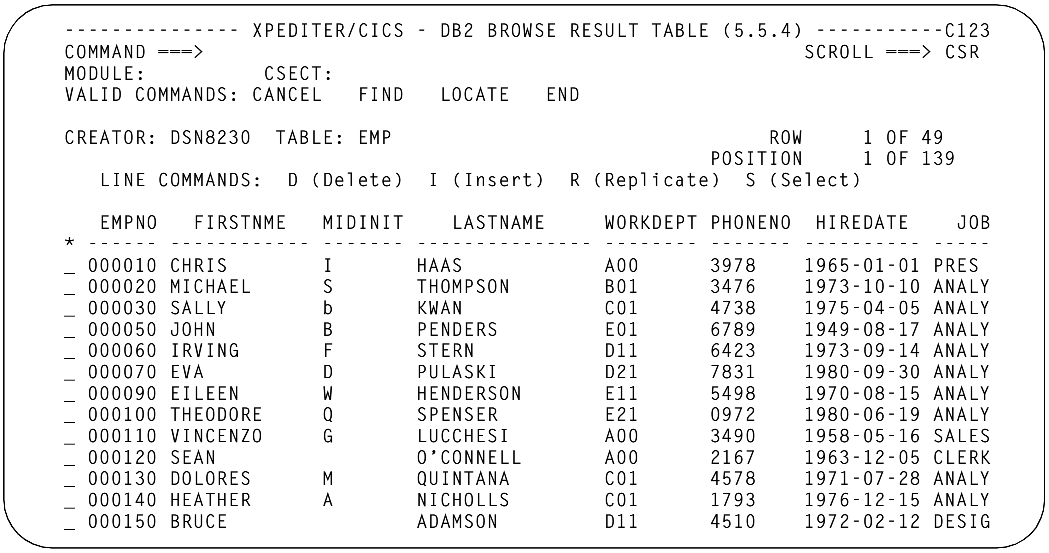
```
------------------------- XPEDITER/CICS - HELP TEXT ---------------------C123
COMMAND ===>                                                SCROLL ===> CSR
MODULE:             ******** HIT PF1 AGAIN FOR HELP ON USING TUTORIALS ********


                 MESSAGE "INTEGER TO DECIMAL CONVERSION ERROR"

   An attempt to convert an integer value to decimal value would result
   in a conversion error because the integer is either too small or too
   large for the scale of the decimal receiving field.

   Examples:


     IF A is defined as DECIMAL(3,0)

       A = -1000  or  A = 28325   fails because the range of valid values
                                  for A would be -999 to +999.


     IF A is defined as DECIMAL(5,3)

       A = -1000  or  A = 28325   fails because the range of valid values
                                  for A would be -99.999 to +99.999.
```

Use the information displayed on the Help screen to determine the source of the error message.

6. Press PF3 (END) to return to the DB2 Edit Result Table Row screen (5.5.5).
7. Position the cursor over the incorrect data and fix the error. Press Enter.
8. Press PF3 (END) to return to the DB2 Browse Result Table screen (5.5.4) (Figure 14-7).

   The changes you made are displayed on this screen. You can use the CANCEL command to cancel the changes. Changes are committed when the END command is used to exit this screen.

Figure 14-7. DB2 Browse Result Table Screen (5.5.4)

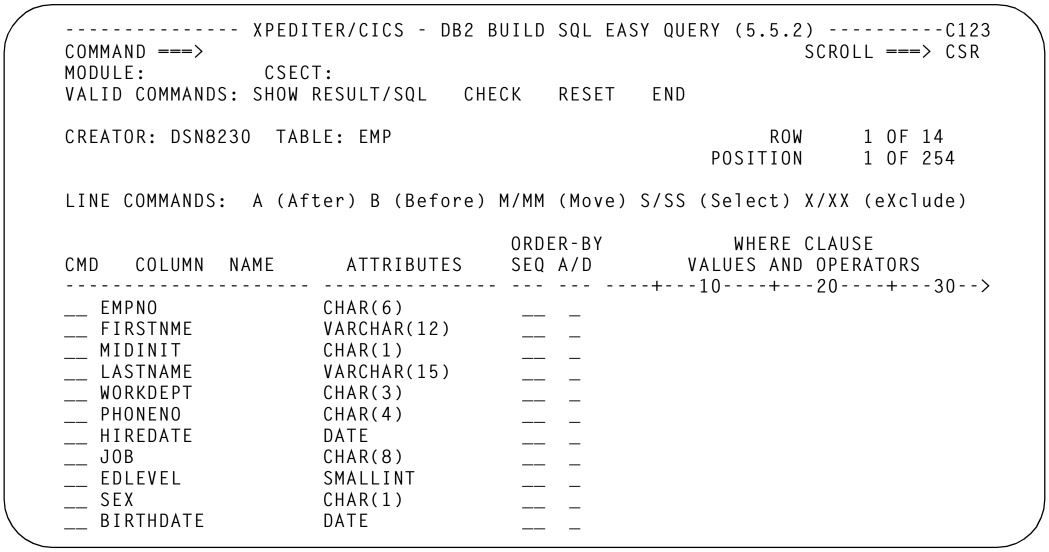


```
-------------- XPEDITER/CICS - DB2 BROWSE RESULT TABLE (5.5.4) ----------C123
COMMAND ===>                                                SCROLL ===> CSR
MODULE:          CSECT:
VALID COMMANDS: CANCEL   FIND   LOCATE   END

CREATOR: DSN8230  TABLE: EMP                                 ROW     1 OF 49
                                                        POSITION     1 OF 139
   LINE COMMANDS:  D (Delete)  I (Insert)  R (Replicate)  S (Select)

   EMPNO   FIRSTNME   MIDINIT    LASTNAME     WORKDEPT PHONENO  HIREDATE    JOB
* ------ ------------ ------- --------------- -------- ------- ---------- -----
_ 000010 CHRIS        I       HAAS            A00      3978    1965-01-01 PRES
_ 000020 MICHAEL      S       THOMPSON        B01      3476    1973-10-10 ANALY
_ 000030 SALLY        b       KWAN            C01      4738    1975-04-05 ANALY
_ 000050 JOHN         B       PENDERS         E01      6789    1949-08-17 ANALY
_ 000060 IRVING       F       STERN           D11      6423    1973-09-14 ANALY
_ 000070 EVA          D       PULASKI         D21      7831    1980-09-30 ANALY
_ 000090 EILEEN       W       HENDERSON       E11      5498    1970-08-15 ANALY
_ 000100 THEODORE     Q       SPENSER         E21      0972    1980-06-19 ANALY
_ 000110 VINCENZO     G       LUCCHESI        A00      3490    1958-05-16 SALES
_ 000120 SEAN                 O'CONNELL       A00      2167    1963-12-05 CLERK
_ 000130 DOLORES      M       QUINTANA        C01      4578    1971-07-28 ANALY
_ 000140 HEATHER      A       NICHOLLS        C01      1793    1976-12-15 ANALY
_ 000150 BRUCE                ADAMSON         D11      4510    1972-02-12 DESIG
```

Building an SQL Easy Query

1. Return to the DB2 Table/View List screen by typing **=5.5.1** in any Xpediter/CICS COMMAND field and pressing Enter.
2. Type the **Q** line command in the CMD field next to a table and press Enter. The DB2 Build SQL Easy Query screen (5.5.2) appears (Figure 14-8).

Figure 14-8. DB2 Build SQL Easy Query Screen (5.5.2)

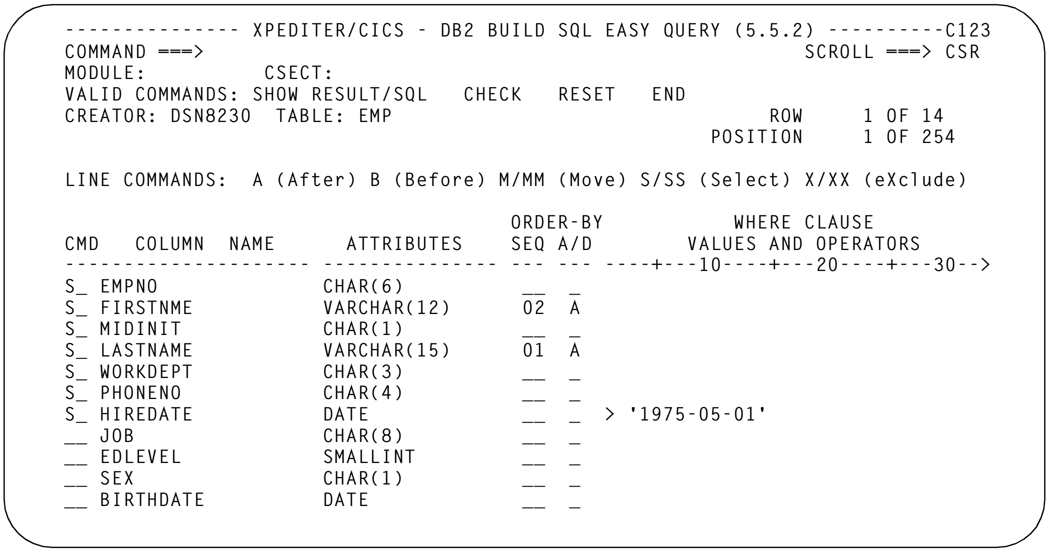


```
--------------- XPEDITER/CICS - DB2 BUILD SQL EASY QUERY (5.5.2) ----------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
VALID COMMANDS: SHOW RESULT/SQL   CHECK   RESET   END

CREATOR: DSN8230  TABLE: EMP                                ROW     1 OF 14
                                                       POSITION     1 OF 254

LINE COMMANDS:  A (After) B (Before) M/MM (Move) S/SS (Select) X/XX (eXclude)

                                      ORDER-BY           WHERE CLAUSE
CMD   COLUMN  NAME      ATTRIBUTES    SEQ A/D        VALUES AND OPERATORS
--------------------- --------------- --- --- ----+---10----+---20----+---30-->
__ EMPNO              CHAR(6)          __  _
__ FIRSTNME           VARCHAR(12)      __  _
__ MIDINIT            CHAR(1)          __  _
__ LASTNAME           VARCHAR(15)      __  _
__ WORKDEPT           CHAR(3)          __  _
__ PHONENO            CHAR(4)          __  _
__ HIREDATE           DATE             __  _
__ JOB                CHAR(8)          __  _
__ EDLEVEL            SMALLINT         __  _
__ SEX                CHAR(1)          __  _
__ BIRTHDATE          DATE             __  _
```

This ISPF-like screen lets you select the fields to be used in the SQL call. You can select the columns to display, the left-to-right order of the columns to be displayed, the row selection using WHERE clauses, and the sort sequence using ORDER-BY clauses.

3. Select desired columns by typing an **S** to the left of each column.
4. Type a two-character number in the ORDER-BY SEQ field to designate the order in which you would like the columns to be sorted.
5. Type **A** or **D** in the ORDER-BY A/D field to specify the ascending or descending sort sequence for this column.
6. Type a WHERE clause in the WHERE CLAUSE field. In the example shown in Figure 14-9, we entered a clause for the HIREDATE, so only those rows with a hire date after May 1, 1975 appear in the result table.
7. Press Enter to process the selection criteria.

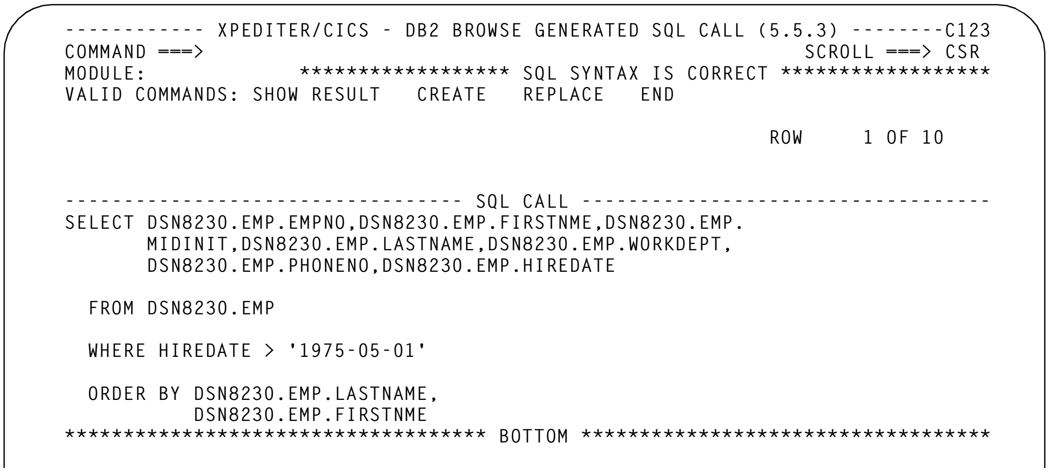
Figure 14-9. DB2 Build SQL Easy Query Screen (5.5.2)



```
-------------- XPEDITER/CICS - DB2 BUILD SQL EASY QUERY (5.5.2) ----------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
VALID COMMANDS: SHOW RESULT/SQL   CHECK   RESET   END
CREATOR: DSN8230  TABLE: EMP                                ROW     1 OF 14
                                                       POSITION     1 OF 254

LINE COMMANDS:  A (After) B (Before) M/MM (Move) S/SS (Select) X/XX (eXclude)

                                      ORDER-BY           WHERE CLAUSE
CMD   COLUMN  NAME      ATTRIBUTES    SEQ A/D        VALUES AND OPERATORS
-------------------- --------------- --- --- ----+---10----+---20----+---30-->
S_ EMPNO              CHAR(6)          __  _
S_ FIRSTNME           VARCHAR(12)      02  A
S_ MIDINIT            CHAR(1)          __  _
S_ LASTNAME           VARCHAR(15)      01  A
S_ WORKDEPT           CHAR(3)          __  _
S_ PHONENO            CHAR(4)          __  _
S_ HIREDATE           DATE             __  _  > '1975-05-01'
__ JOB                CHAR(8)          __  _
__ EDLEVEL            SMALLINT         __  _
__ SEX                CHAR(1)          __  _
__ BIRTHDATE          DATE             __  _
```

8. Type **SHOW SQL** in the COMMAND field and press Enter. Xpediter/CICS generates an SQL call, and the DB2 Browse Generated SQL Call screen (5.5.3) (Figure 14-10) displays the actual SQL statement.

Figure 14-10. DB2 Browse Generated SQL Call Screen (5.5.3)

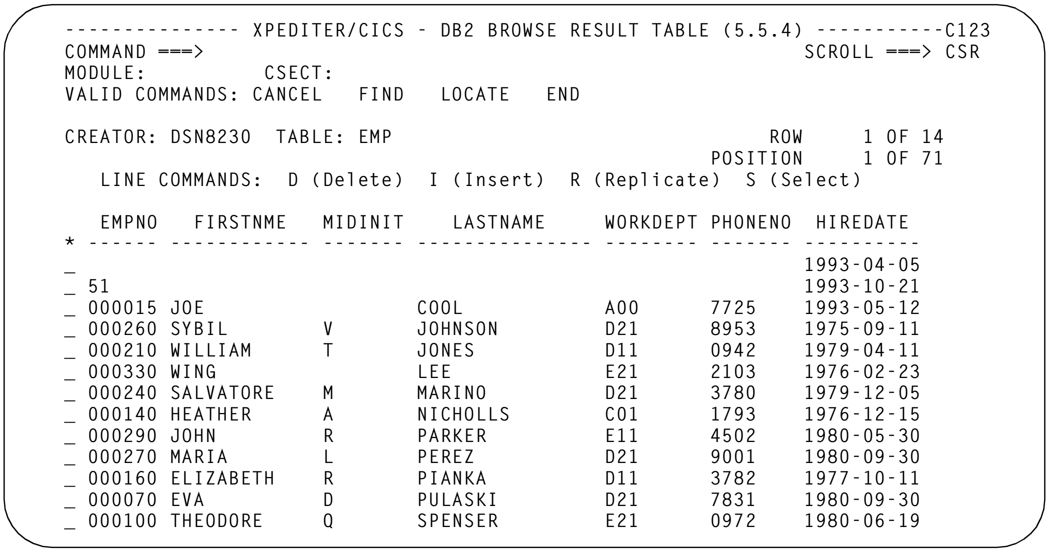


```
----------- XPEDITER/CICS - DB2 BROWSE GENERATED SQL CALL (5.5.3) --------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:             ****************** SQL SYNTAX IS CORRECT ******************
VALID COMMANDS: SHOW RESULT   CREATE   REPLACE   END

                                                            ROW     1 OF 10


---------------------------------- SQL CALL ------------------------------------
SELECT DSN8230.EMP.EMPNO,DSN8230.EMP.FIRSTNME,DSN8230.EMP.
       MIDINIT,DSN8230.EMP.LASTNAME,DSN8230.EMP.WORKDEPT,
       DSN8230.EMP.PHONENO,DSN8230.EMP.HIREDATE

  FROM DSN8230.EMP

  WHERE HIREDATE > '1975-05-01'

  ORDER BY DSN8230.EMP.LASTNAME,
           DSN8230.EMP.FIRSTNME
************************************ BOTTOM ***********************************
```

You can save the SQL call in the Xpediter/CICS SQL transfer file to be printed or included in a program. Each SQL call is stored as a single record in this VSAM file with the name you supply as an operand on the CREATE or REPLACE command. In this way, you can use Xpediter/CICS to generate and test SQL calls before your program is written, then include them in your program code. For more information, see Chapter 6, “DB2 Format Utility,” in the *Xpediter/CICS Installation Guide*.

9. Type **SHOW RESULT** in the COMMAND field and press Enter to see the result table generated by this call on the DB2 Browse Result Table screen (5.5.4) (Figure 14-11).

Figure 14-11. DB2 Browse Result Table Screen (5.5.4)

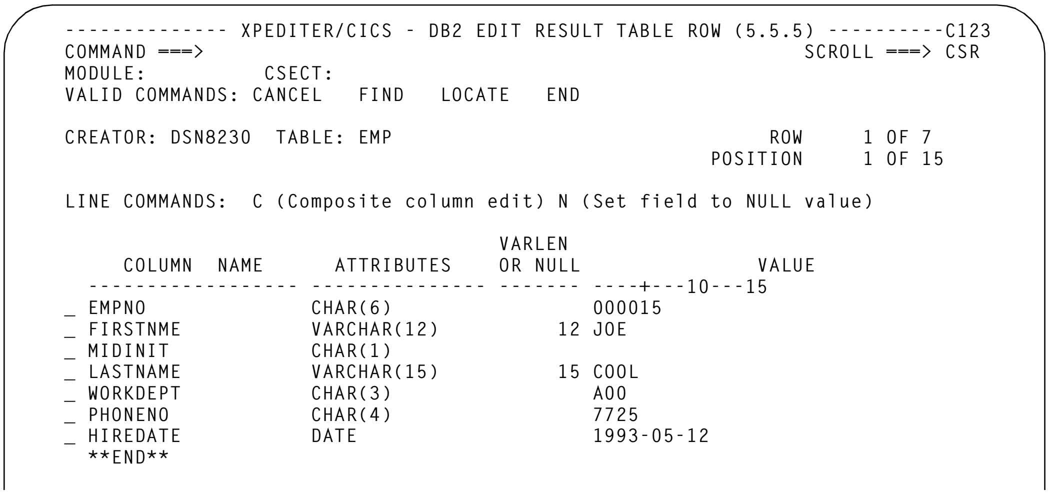


```
--------------- XPEDITER/CICS - DB2 BROWSE RESULT TABLE (5.5.4) -----------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
VALID COMMANDS: CANCEL   FIND   LOCATE   END

CREATOR: DSN8230  TABLE: EMP                                ROW     1 OF 14
                                                       POSITION     1 OF 71
   LINE COMMANDS:  D (Delete)  I (Insert)  R (Replicate)  S (Select)

   EMPNO   FIRSTNME   MIDINIT    LASTNAME     WORKDEPT PHONENO  HIREDATE
* ------ ------------ ------- --------------- -------- ------- ----------
_                                                              1993-04-05
_ 51                                                           1993-10-21
_ 000015 JOE                  COOL            A00      7725    1993-05-12
_ 000260 SYBIL        V       JOHNSON         D21      8953    1975-09-11
_ 000210 WILLIAM      T       JONES           D11      0942    1979-04-11
_ 000330 WING                 LEE             E21      2103    1976-02-23
_ 000240 SALVATORE    M       MARINO          D21      3780    1979-12-05
_ 000140 HEATHER      A       NICHOLLS        C01      1793    1976-12-15
_ 000290 JOHN         R       PARKER          E11      4502    1980-05-30
_ 000270 MARIA        L       PEREZ           D21      9001    1980-09-30
_ 000160 ELIZABETH    R       PIANKA          D11      3782    1977-10-11
_ 000070 EVA          D       PULASKI         D21      7831    1980-09-30
_ 000100 THEODORE     Q       SPENSER         E21      0972    1980-06-19
```

This is the same screen that was discussed in "Editing a Row" on page 14-4. Rows can be selected from this table for updating.

10. Type the **R** line command next to a row and press Enter to replicate this row.
11. Type the **S** line command next to the new row and press Enter. The DB2 Edit Result Table Row screen (5.5.5) appears (Figure 14-12).
12. Type new information in the VALUE field of this screen and press Enter. In the example shown here, JOE COOL will be changed to RALPH COOL.
13. The changes are updated in the new row.

Figure 14-12. DB2 Edit Result Table Row Screen (5.5.5)

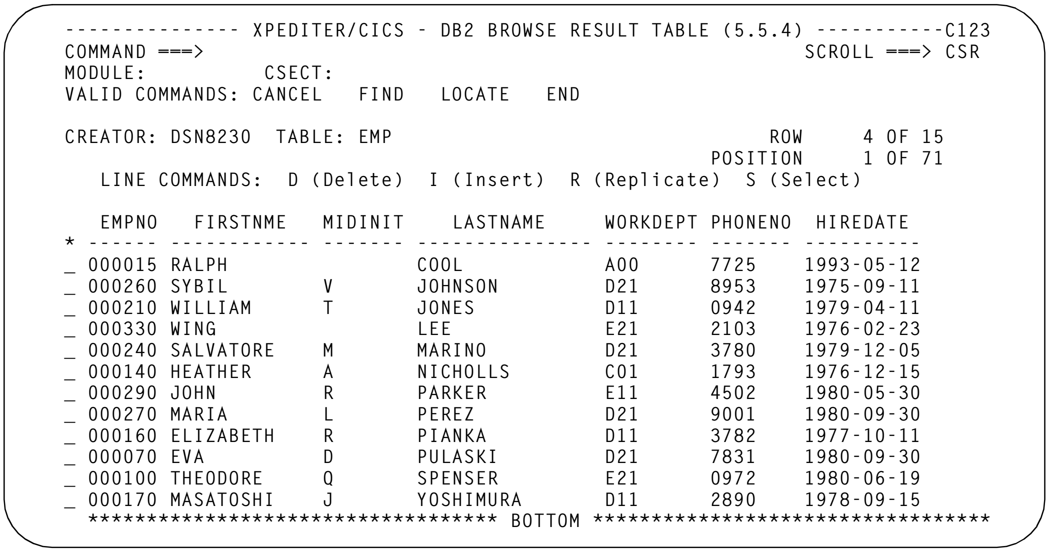


```
-------------- XPEDITER/CICS - DB2 EDIT RESULT TABLE ROW (5.5.5) ----------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
VALID COMMANDS: CANCEL   FIND   LOCATE   END

CREATOR: DSN8230  TABLE: EMP                                ROW     1 OF 7
                                                       POSITION     1 OF 15

LINE COMMANDS:  C (Composite column edit) N (Set field to NULL value)

                                    VARLEN
     COLUMN  NAME      ATTRIBUTES    OR NULL              VALUE
  ----------------- --------------- ------- ----+---10---15
_ EMPNO             CHAR(6)                 000015
_ FIRSTNME          VARCHAR(12)          12 JOE
_ MIDINIT           CHAR(1)
_ LASTNAME          VARCHAR(15)          15 COOL
_ WORKDEPT          CHAR(3)                 A00
_ PHONENO           CHAR(4)                 7725
_ HIREDATE          DATE                    1993-05-12
  **END**
```

14. Press PF3 (END) to see the DB2 Browse Result Table screen (5.5.4) Figure 14-13.

    Remember that unless the CANCEL command is used, changes are committed when you leave the DB2 Browse Result Table screen (5.5.4).

Figure 14-13. DB2 Browse Result Table Screen (5.5.4)

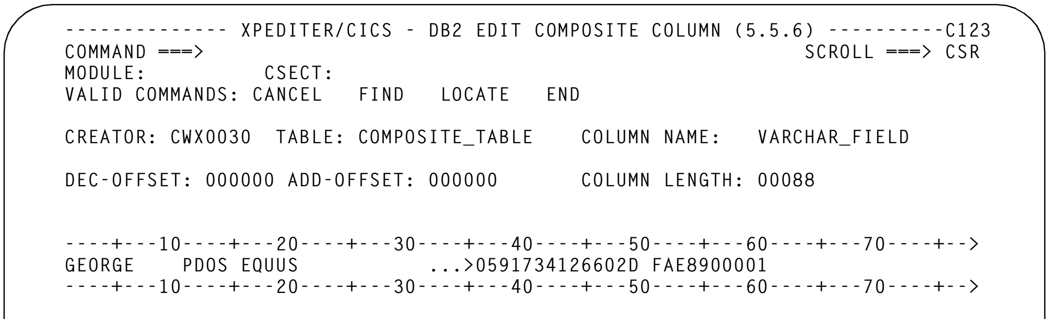


```
--------------- XPEDITER/CICS - DB2 BROWSE RESULT TABLE (5.5.4) -----------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
VALID COMMANDS: CANCEL   FIND   LOCATE   END

CREATOR: DSN8230  TABLE: EMP                                ROW     4 OF 15
                                                       POSITION     1 OF 71
   LINE COMMANDS:  D (Delete)  I (Insert)  R (Replicate)  S (Select)

   EMPNO   FIRSTNME   MIDINIT    LASTNAME     WORKDEPT PHONENO  HIREDATE
* ------ ------------ ------- --------------- -------- ------- ----------
_ 000015 RALPH                COOL            A00      7725    1993-05-12
_ 000260 SYBIL        V       JOHNSON         D21      8953    1975-09-11
_ 000210 WILLIAM      T       JONES           D11      0942    1979-04-11
_ 000330 WING                 LEE             E21      2103    1976-02-23
_ 000240 SALVATORE    M       MARINO          D21      3780    1979-12-05
_ 000140 HEATHER      A       NICHOLLS        C01      1793    1976-12-15
_ 000290 JOHN         R       PARKER          E11      4502    1980-05-30
_ 000270 MARIA        L       PEREZ           D21      9001    1980-09-30
_ 000160 ELIZABETH    R       PIANKA          D11      3782    1977-10-11
_ 000070 EVA          D       PULASKI         D21      7831    1980-09-30
_ 000100 THEODORE     Q       SPENSER         E21      0972    1980-06-19
_ 000170 MASATOSHI    J       YOSHIMURA       D11      2890    1978-09-15
  ********************************** BOTTOM *********************************
```

Editing a Column

The DB2 Edit Composite Column screen (5.5.6) (Figure 14-14) lets you edit data in a DB2 result table column. This screen is accessed by using the C line command from the DB2 Edit Result Table Row screen (5.5.5). Composite columns are defined in Xpediter/CICS as DATE, TIME, TIMESTAMP, CHARACTER, or GRAPHIC columns that are redefined by a user in an application program. Use this support when the 300-column limit is too restrictive, or if you have kanji data in a column.

Figure 14-14. DB2 Edit Composite Column Screen (5.5.6)



```
-------------- XPEDITER/CICS - DB2 EDIT COMPOSITE COLUMN (5.5.6) ----------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
VALID COMMANDS: CANCEL   FIND   LOCATE   END

CREATOR: CWX0030  TABLE: COMPOSITE_TABLE    COLUMN NAME:   VARCHAR_FIELD

DEC-OFFSET: 000000 ADD-OFFSET: 000000       COLUMN LENGTH: 00088


----+---10----+---20----+---30----+---40----+---50----+---60----+---70----+-->
GEORGE    PDOS EQUUS          ...>0591734126602D FAE8900001
----+---10----+---20----+---30----+---40----+---50----+---60----+---70----+-->
```

DB2 Long Identifier Considerations

IBM DB2 UDB V8 for z/OS introduced many changes to the DB2 product, including the introduction of long identifier fields in DB2 V8 New Function Mode (NFM). The identifier lengths that affect the Xpediter/CICS DB2 File Utility are as follows:

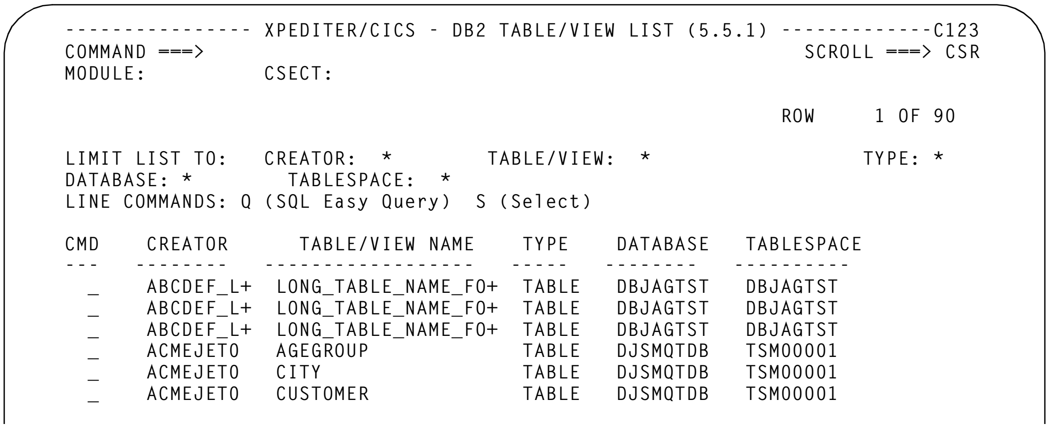
- The permissible length of the table creator name has increased from 8 to 128 bytes.
- The permissible length of the table/view name has increased from 18 to 128 bytes.
- The permissible length of the column name has increased from 18 to 30 bytes.

The Xpediter/CICS DB2 File Utility provides all of the functionality of the previous releases of the DB2 File Utility, including support for DB2 and long identifiers. Because of screen limitations, however, the maximum display sizes for the creator, table name, and column name fields are limited to 8, 18, and 18 bytes respectively in Xpediter/CICS. If a DB2 field exceeds these lengths, the field is truncated for display (only), and a plus character "+" is appended to the right of that field. For example, a creator name of CREATOR_NAME_IS_LONGER_THAN_V7 is truncated to 8 bytes and displayed as CREATOR_+, a table name of TABLE_NAME_IS_LONGER_THAN_V7 is truncated to 18 bytes and displayed as TABLE_NAME_IS_LONG+, and a column name of COLUMN_IS_LONGER_THAN_V7 is truncated to 18 bytes and displayed as COLUMN_IS_LONGER_T+. **The full length of these fields is only displayed on the DB2 BROWSE GENERATED SQL CALL (5.5.3) screen.**

Following are examples of three DB2 File Utilty screens showing the result of displaying the DB2 long identifiers in the File Utility.

Figure 14-15 shows an example of the DB2 TABLE/VIEW LIST (5.5.1) screen. In the first row of this display the long creator ABCDEF_LONG_CREATOR_TEST has been truncated and is displayed as eight bytes ABCDEF_L and a plus sign (+) is appended to the right of the field. Likewise the table name of LONG_TABLE_NAME_FOR__TESTING_DCLGEN has been truncated to eighteen bytes and a plus sign appended.

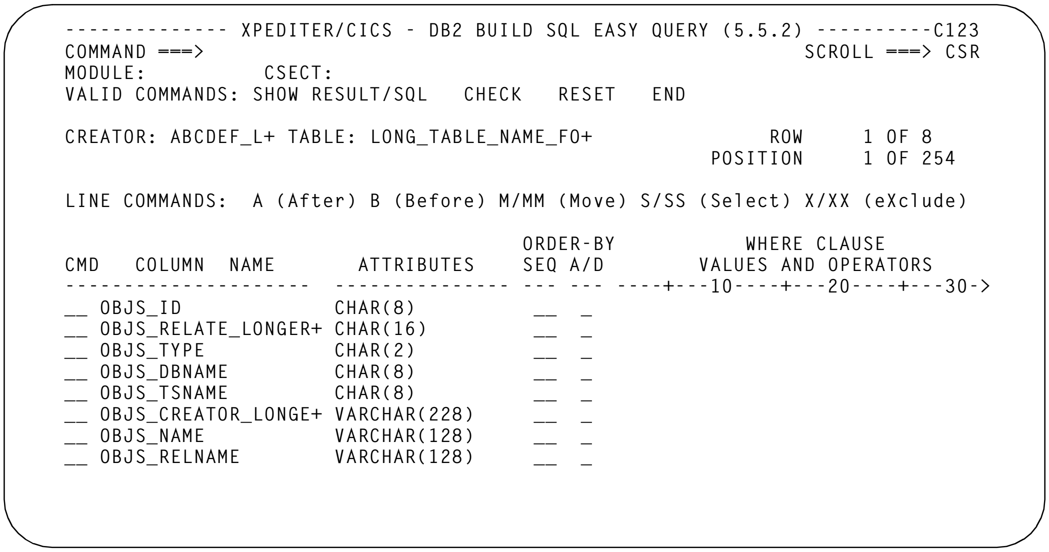
Figure 14-15. DB2 Table/View List



```
--------------- XPEDITER/CICS - DB2 TABLE/VIEW LIST (5.5.1) -------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE:         CSECT:

                                                            ROW     1 OF 90

LIMIT LIST TO:  CREATOR:  *        TABLE/VIEW:  *                  TYPE: *
DATABASE: *       TABLESPACE:  *
LINE COMMANDS: Q (SQL Easy Query)  S (Select)

CMD   CREATOR      TABLE/VIEW NAME    TYPE    DATABASE   TABLESPACE
---   --------  ------------------   -----   --------   ----------
  _   ABCDEF_L+  LONG_TABLE_NAME_FO+  TABLE   DBJAGTST   DBJAGTST
  _   ABCDEF_L+  LONG_TABLE_NAME_FO+  TABLE   DBJAGTST   DBJAGTST
  _   ABCDEF_L+  LONG_TABLE_NAME_FO+  TABLE   DBJAGTST   DBJAGTST
  _   ACMEJETO   AGEGROUP             TABLE   DJSMQTDB   TSM00001
  _   ACMEJETO   CITY                 TABLE   DJSMQTDB   TSM00001
  _   ACMEJETO   CUSTOMER             TABLE   DJSMQTDB   TSM00001
```

Typing a Q in the command field of the first row and pressing Enter causes the DB2 BUILD SQL EASY QUERY (5.5.2) screen in Figure 14-16 to be displayed. In the heading both the CREATOR: and TABLE: data has been truncated for display and a plus sign has been appended to the right of each field. The long column name OBJS_RELATE_LONGER_CL_NAME has been truncated to eighteen bytes and a plus sign has been appended to the right of the field.

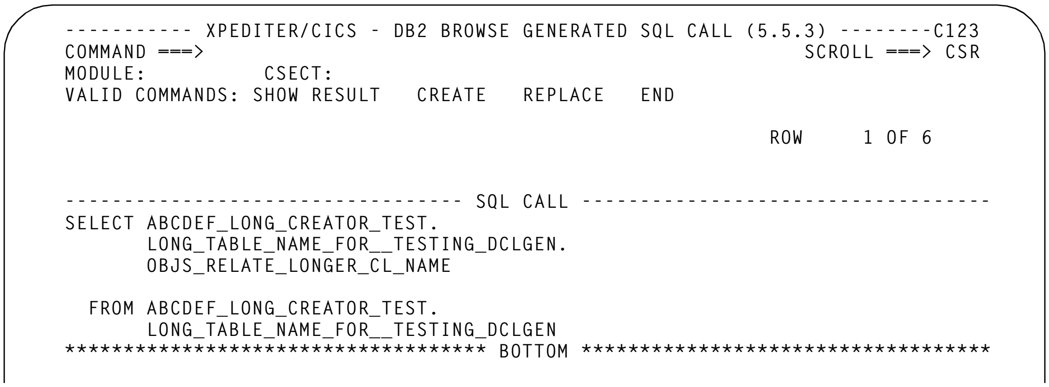
Figure 14-16. DB2 Build SQL Easy Query



```
------------- XPEDITER/CICS - DB2 BUILD SQL EASY QUERY (5.5.2) ---------C123
COMMAND ===>                                                 SCROLL ===> CSR
MODULE:          CSECT:
VALID COMMANDS: SHOW RESULT/SQL   CHECK   RESET   END

CREATOR: ABCDEF_L+ TABLE: LONG_TABLE_NAME_FO+              ROW     1 OF 8
                                                      POSITION     1 OF 254

LINE COMMANDS:  A (After) B (Before) M/MM (Move) S/SS (Select) X/XX (eXclude)

                                       ORDER-BY          WHERE CLAUSE
CMD   COLUMN  NAME       ATTRIBUTES    SEQ A/D       VALUES AND OPERATORS
--------------------  --------------- --- --- ----+---10----+---20----+---30->
__ OBJS_ID            CHAR(8)          __  _
__ OBJS_RELATE_LONGER+ CHAR(16)        __  _
__ OBJS_TYPE          CHAR(2)          __  _
__ OBJS_DBNAME        CHAR(8)          __  _
__ OBJS_TSNAME        CHAR(8)          __  _
__ OBJS_CREATOR_LONGE+ VARCHAR(228)    __  _
__ OBJS_NAME          VARCHAR(128)     __  _
__ OBJS_RELNAME       VARCHAR(128)     __  _
```

Entering the primary command SHOW SQL, typing an S in the command field for the column name OBJS_RELATE_LONGER, and pressing Enter causes the DB2 BROWSE GENERATED SQL CALL (5.5.3) screen in Figure 14-17 to be displayed. Notice on this screen that the full lengths of the creator, table name, and column are displayed in the generated SQL call.

Figure 14-17. DB2 Browse Generated SQL Call



```
---------- XPEDITER/CICS - DB2 BROWSE GENERATED SQL CALL (5.5.3) --------C123
COMMAND ===>                                                 SCROLL ===> CSR
MODULE:          CSECT:
VALID COMMANDS: SHOW RESULT   CREATE   REPLACE   END

                                                           ROW     1 OF 6


---------------------------------- SQL CALL -----------------------------------
SELECT ABCDEF_LONG_CREATOR_TEST.
       LONG_TABLE_NAME_FOR__TESTING_DCLGEN.
       OBJS_RELATE_LONGER_CL_NAME

  FROM ABCDEF_LONG_CREATOR_TEST.
       LONG_TABLE_NAME_FOR__TESTING_DCLGEN
************************************ BOTTOM ***********************************
```

Debugging DB2 Programs

This section discusses the special facilities available to DB2 programmers, including setting breakpoints and keeps, interpreting abends and SQL codes, and accessing DB2 storage.

Setting Breakpoints in SQL Code

Your online source listing displays both commented-out SQL commands and the DB2 translated code. As a result, Xpediter/CICS lets you set or delete breakpoints in all generated instructions. In addition, Xpediter/CICS allows you to globally set breakpoints on all SQL calls.

- To set breakpoints before every SQL statement, enter **BEFORE ALL SQL** in the COMMAND field and press Enter. Xpediter/CICS dynamically sets before breakpoints on every SQL statement or call to DSNHLI.

  You can also set breakpoints after EXEC SQL statements and counts of EXEC SQL statements. Breakpoints can be set on all returning SQL calls, or counts can be set to help in SQL analysis.
- DB2 breakpoints are further qualified by specifying an SQL call type. For example, **COUNT ALL SQL UPDATE** sets counters only on EXEC SQL UPDATE calls. No other calls are counted. Of course, you can set or delete individual breakpoints using the BEFORE, AFTER, RUNTO, COUNT, and DELETE commands.

Setting Keeps on DB2-Specific Data

In Xpediter/CICS, the KEEP command is used to continuously view a data field on the Source Listing screen (2.L). You can display various DB2 data items, such as working storage items, DFHCOMM fields, indices, and DB2 fields. You can add any DB2 specific data item, such as SQLCODE or SQLERRM, to the keep window, as long as it is defined to your program.

To display the current SQLCODE value, enter **KEEP SQLCODE** in the COMMAND field. Once displayed, you can modify data by replacing the contents of the field. This is an excellent way to test IF logic after an SQL call.

Interpreting DSNC Abends and SQL Codes

Xpediter/CICS automatically traps all encountered abends, including DB2 DSNC abends. Xpediter/CICS recognizes DSNC reason codes and treats them like any other abend. It intercepts the abend before it takes place and returns control to you.

DSNC Abends

For example, if your DB2 program abends with a DSNC AEY9 abend code, Xpediter/CICS will:

- Intercept the abend
- Format the Source Listing screen
- Point to the offending call
- Flag the abend code as an AEY9.

If CICS Abend-AID is installed, additional DB2 information is available. Typing =7.1 in the COMMAND field and pressing Enter allows you to jump directly into CICS Abend-AID to diagnose an AEY9 as a call to DB2 prior to activation.

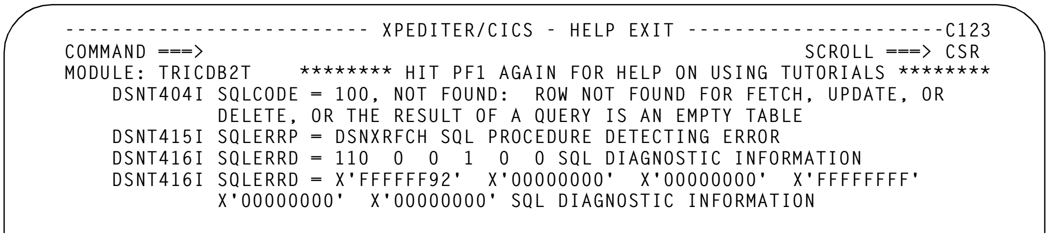
DSNC abend recognition is especially useful in test regions prone to DB2 attachment-facility errors. This Xpediter feature is designed to assist both the DBA and the DB2 application programmer.

SQL Codes

Programmers commonly complain about cryptic return codes. This is also true for DB2 SQL codes. Xpediter/CICS has SQL support within its Help facility.

Enter **HELP SQLERROR**, **HELP SQLCODE**, or **HELP SQL** to access timely SQL diagnostics and warnings of the last SQL statement executed. Figure 14-18 is an example of help after DB2 returned a 100 SQLCODE.

Figure 14-18. Help Exit Screen



```
------------------------- XPEDITER/CICS - HELP EXIT ---------------------C123
COMMAND ===>                                                 SCROLL ===> CSR
MODULE: TRICDB2T    ******** HIT PF1 AGAIN FOR HELP ON USING TUTORIALS ********
    DSNT404I SQLCODE = 100, NOT FOUND:  ROW NOT FOUND FOR FETCH, UPDATE, OR
             DELETE, OR THE RESULT OF A QUERY IS AN EMPTY TABLE
    DSNT415I SQLERRP = DSNXRFCH SQL PROCEDURE DETECTING ERROR
    DSNT416I SQLERRD = 110  0  0  1  0  0 SQL DIAGNOSTIC INFORMATION
    DSNT416I SQLERRD = X'FFFFFF92'  X'00000000'  X'00000000'  X'FFFFFFFF'
             X'00000000'  X'00000000' SQL DIAGNOSTIC INFORMATION
```

Accessing DB2 Storage

Xpediter/CICS provides extensive access to any CICS table or control area. These areas are accessed in hexadecimal using the Memory Display screen (2.2), or mapped to a DSECT of the current IBM data area description using the DSECTs screen (2.D). Both screens allow keyword access.

Five DB2 related keywords are available for use in the TABLE/AREA field of the Memory Display (2.2) and DSECTs (2.D) screens. These commands are useful in diagnosing DB2 programs with the following storage problems:

RCT (DB2 Resource Control Table): Identifies the plan name, thread, and TCB information.

Since the RCT is no longer available, if RCT is specified, the CICS Resources screen (2.R) is displayed showing information for the DB2CONN entry.

SQLCA (SQL Communications Area): Identifies the SQL return code, error diagnostics, and warning indicators of the last SQL statement executed.

SQLDA (SQL Descriptor Area): Provides a pointer to the data received by a SELECT statement in a dynamic SQL call.

PLIST (DB2 Parameter List): List of parameters passed to the DB2 call generated by an EXEC SQL program statement.

CLOT (CICS Life of Task): Provides the DB2 connection authorization ID, and various error codes associated with the task.

Note: All of the above keywords (except RCT) can only be used at a break or abend with DB2 active.

Chapter 15.
Using Xpediter/CICS with MQ

Xpediter/CICS provides support for IBM's WebSphere MQ (formerly MQSeries) messaging manager. In addition to the extensive interactive debugging facilities available to all CICS programs, special facilities have been created to meet the needs of the MQ programmer.

Note: The demonstrations in this chapter should be performed in Xpediter's standard operating mode — ***not*** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

Using the MQ File Utility

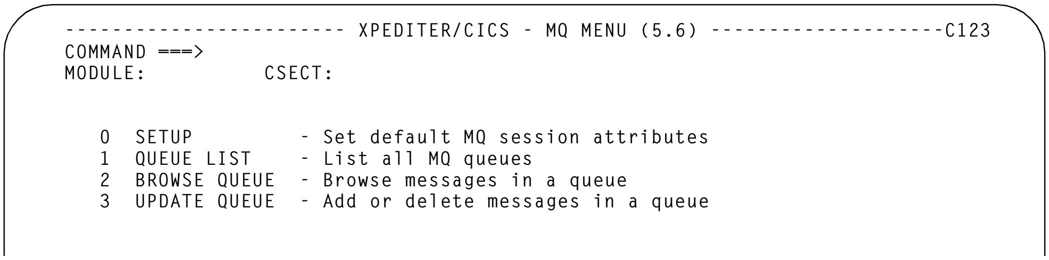
Xpediter/CICS provides a list of MQ queues available to the CICS system and allows you to browse messages on local queues. You can also add and delete messages on an MQ queue. For more information, see the descriptions of the DELETE, GETPUT, and PUT commands in the *Xpediter/CICS Reference Manual*. The MQ File Utility in Xpediter/CICS honors all MQ security rules. You can use the MQ File Utility only with the queues you have authority to access.

Note: Messages created by a sample application were used to generate the screens shown in this chapter. Since the data you access will be different, your screens will vary from those shown. Use this chapter simply as a model of how to access your queues.

Accessing a List of MQ Queues

1. From a blank CICS screen, type **XPED 5.6** and press Enter to display the MQ Menu screen (5.6) shown in Figure 15-1.

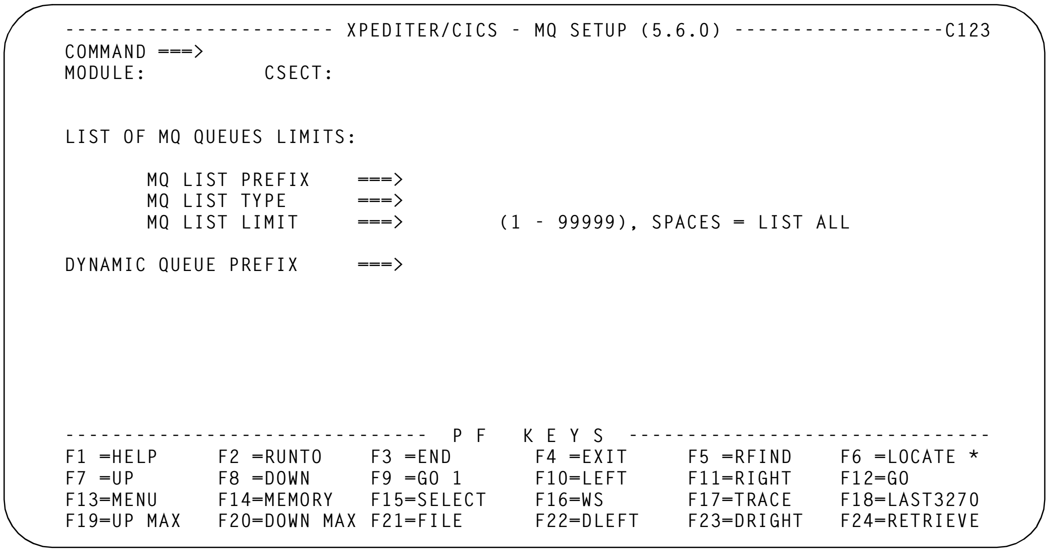
Figure 15-1. MQ Menu (5.6)



```
----------------------- XPEDITER/CICS - MQ MENU (5.6) --------------------C123
COMMAND ===>
MODULE:          CSECT:


   0  SETUP         - Set default MQ session attributes
   1  QUEUE LIST    - List all MQ queues
   2  BROWSE QUEUE  - Browse messages in a queue
   3  UPDATE QUEUE  - Add or delete messages in a queue
```

2. Type **0** and press Enter to display the MQ SETUP screen (5.6.0) shown in Figure 15-2.

Figure 15-2. MQ Setup (5.6.0)

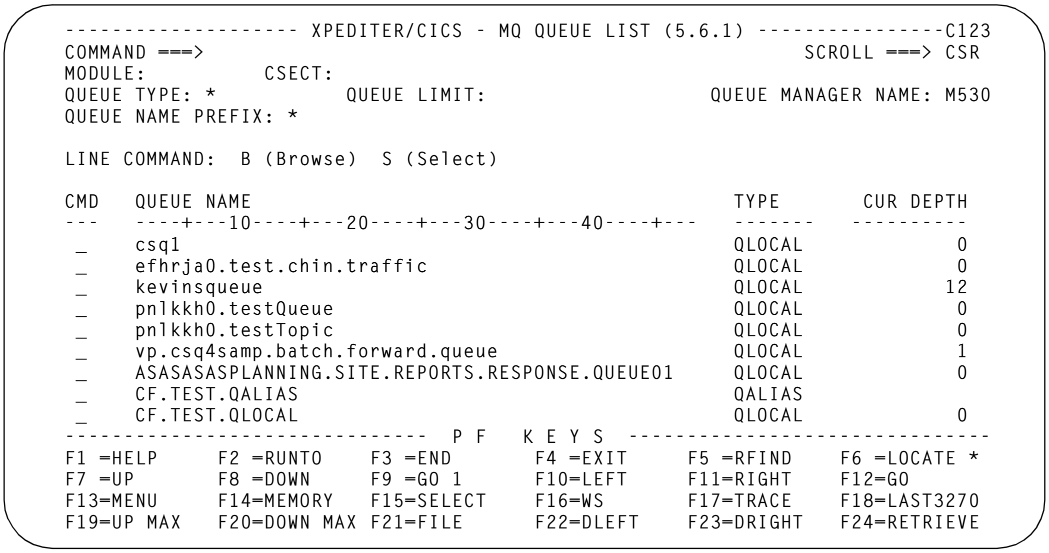


```
----------------------- XPEDITER/CICS - MQ SETUP (5.6.0) ------------------C123
COMMAND ===>
MODULE:          CSECT:


LIST OF MQ QUEUES LIMITS:

       MQ LIST PREFIX    ===>
       MQ LIST TYPE      ===>
       MQ LIST LIMIT     ===>        (1 - 99999), SPACES = LIST ALL

DYNAMIC QUEUE PREFIX     ===>






-------------------------------  P F   K E Y S  -------------------------------
F1 =HELP     F2 =RUNTO    F3 =END       F4 =EXIT     F5 =RFIND    F6 =LOCATE *
F7 =UP       F8 =DOWN     F9 =GO 1      F10=LEFT     F11=RIGHT    F12=GO
F13=MENU     F14=MEMORY   F15=SELECT    F16=WS       F17=TRACE    F18=LAST3270
F19=UP MAX   F20=DOWN MAX F21=FILE      F22=DLEFT    F23=DRIGHT   F24=RETRIEVE
```

The MQ LIST PREFIX, MQ LIST TYPE, and MQ LIST LIMIT can be customized or established in each of these fields to limit the queues displayed on the MQ Queue List (5.6.1) screen. The DYNAMIC QUEUE PREFIX can also be entered for your session if you need to override your site’s prefix which is found in global. If the Xpediter/CICS profile dataset is used, your overrides are saved for future debugging sessions.

3. Type **1** and press Enter to display the MQ Queue List screen (5.6.1) (Figure 15-3).

Figure 15-3. MQ Queue List (5.6.1)



```
-------------------- XPEDITER/CICS - MQ QUEUE LIST (5.6.1) ----------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE:          CSECT:
QUEUE TYPE: *           QUEUE LIMIT:                  QUEUE MANAGER NAME: M530
QUEUE NAME PREFIX: *

LINE COMMAND:  B (Browse)  S (Select)

CMD   QUEUE NAME                                        TYPE       CUR DEPTH
---   ----+---10----+---20----+---30----+---40----+---   -------   ----------
 _    csq1                                              QLOCAL             0
 _    efhrja0.test.chin.traffic                         QLOCAL             0
 _    kevinsqueue                                       QLOCAL            12
 _    pnlkkh0.testQueue                                 QLOCAL             0
 _    pnlkkh0.testTopic                                 QLOCAL             0
 _    vp.csq4samp.batch.forward.queue                   QLOCAL             1
 _    ASASASASPLANNING.SITE.REPORTS.RESPONSE.QUEUE01    QLOCAL             0
 _    CF.TEST.QALIAS                                    QALIAS
 _    CF.TEST.QLOCAL                                    QLOCAL             0
-------------------------------  P F   K E Y S  -------------------------------
F1 =HELP     F2 =RUNTO    F3 =END       F4 =EXIT     F5 =RFIND    F6 =LOCATE *
F7 =UP       F8 =DOWN     F9 =GO 1      F10=LEFT     F11=RIGHT    F12=GO
F13=MENU     F14=MEMORY   F15=SELECT    F16=WS       F17=TRACE    F18=LAST3270
F19=UP MAX   F20=DOWN MAX F21=FILE      F22=DLEFT    F23=DRIGHT   F24=RETRIEVE
```

The QUEUE TYPE field, QUEUE NAME PREFIX field, and QUEUE LIMIT field are used to limit the queues displayed on this screen.

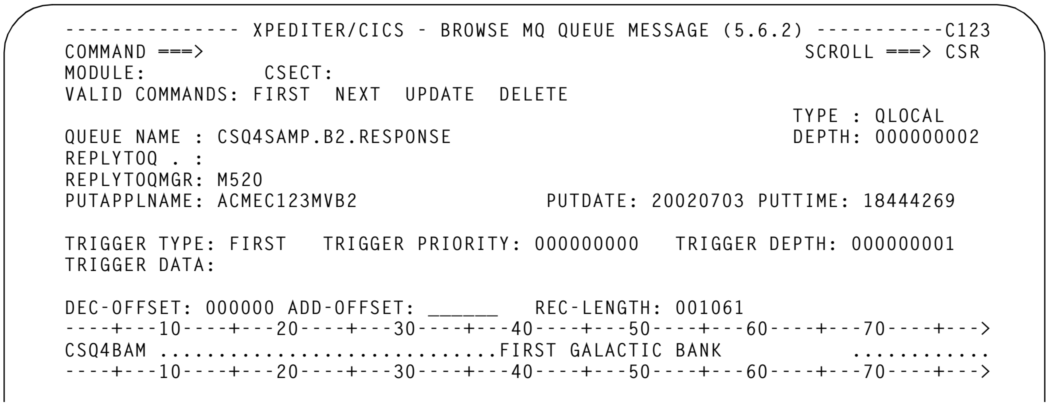
4. To change a value on the screen, type over the existing value and press Enter. Note that the QUEUE NAME PREFIX field is case-sensitive.

Browsing Messages on an MQ Queue

1. On the MQ Queue List screen, locate any queue you are authorized to access that has a non-zero value in the DEPTH column. You may need to scroll to locate an appropriate queue.
2. Type the **B** line command in the CMD field next to the queue and press Enter to transfer to the Browse MQ Queue Message screen (5.6.2) shown in Figure 15-4.

Note: Browsing a message from an initiation queue may cause a trigger event to occur, which may result in a trigger message being generated in the initiation queue.

Figure 15-4. Browse MQ Queue Message Screen (5.6.2)



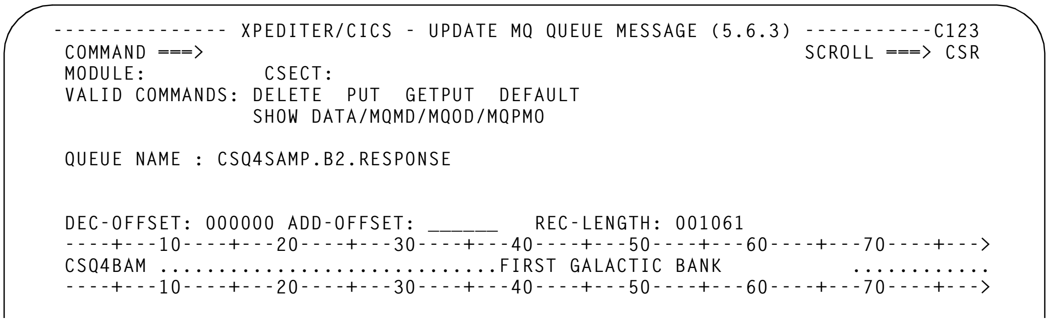
```
--------------- XPEDITER/CICS - BROWSE MQ QUEUE MESSAGE (5.6.2) -----------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:         CSECT:
VALID COMMANDS: FIRST  NEXT  UPDATE  DELETE
                                                              TYPE : QLOCAL
QUEUE NAME : CSQ4SAMP.B2.RESPONSE                             DEPTH: 000000002
REPLYTOQ . :
REPLYTOQMGR: M520
PUTAPPLNAME: ACMEC123MVB2               PUTDATE: 20020703 PUTTIME: 18444269

TRIGGER TYPE: FIRST   TRIGGER PRIORITY: 000000000   TRIGGER DEPTH: 000000001
TRIGGER DATA:

DEC-OFFSET: 000000 ADD-OFFSET: ______   REC-LENGTH: 001061
----+---10----+---20----+---30----+---40----+---50----+---60----+---70----+--->
CSQ4BAM .............................FIRST GALACTIC BANK           ............
----+---10----+---20----+---30----+---40----+---50----+---60----+---70----+--->
```

The Browse MQ Queue Message screen (5.6.2) displays the first message on the selected queue. There are two primary commands available on this screen: NEXT to browse the next message on the queue and FIRST to reposition to the first message on the queue. There may be two additional primary commands available on this screen: UPDATE to transfer to the Update MQ Queue Message screen (5.6.3) and DELETE to delete the currently accessed message.

Adding Messages on an MQ Queue

1. On the Browse MQ Queue Message screen (5.6.2), type the **UPDATE** primary command and press Enter to transfer to the Update MQ Queue Message screen (5.6.3) shown in Figure 15-5.

Figure 15-5. Update MQ Queue Message Screen (5.6.3)



```
--------------- XPEDITER/CICS - UPDATE MQ QUEUE MESSAGE (5.6.3) -----------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:         CSECT:
VALID COMMANDS: DELETE  PUT  GETPUT  DEFAULT
                SHOW DATA/MQMD/MQOD/MQPMO

QUEUE NAME : CSQ4SAMP.B2.RESPONSE


DEC-OFFSET: 000000 ADD-OFFSET: ______   REC-LENGTH: 001061
----+---10----+---20----+---30----+---40----+---50----+---60----+---70----+--->
CSQ4BAM .............................FIRST GALACTIC BANK           ............
----+---10----+---20----+---30----+---40----+---50----+---60----+---70----+--->
```

The Update MQ Queue Message screen (5.6.3) displays the same message as the Browse MQ Queue Message screen (5.6.2).

Debugging MQ Programs

This section discusses the special facilities available to WebSphere MQ programmers, including setting breakpoints and keeps and interpreting MQ completion and reason codes.

Enhanced Traps for MQ Data

Enhanced trap support is also available for MQ. This support allows conditional traps on data in the MQ Message Descriptor (MQMD) or MQ data area.

The traps may be defined using Xpediter's Trap Summary screen (1.6 or 9.6), or a label may be defined on the Define User Labels screen (1.9) or Define System Labels screen (9.9) for use in a conditional trap, breakpoint, or skip. Please see the *Xpediter/CICS Reference Manual* for more details.

Setting Breakpoints at MQ Calls

Xpediter/CICS allows you to globally set breakpoints on all MQ calls. To set breakpoints before every call to MQ, type **BEFORE ALL MQ** in the COMMAND field and press Enter. Xpediter/CICS dynamically sets before breakpoints on every call to MQ. Of course, you can set or delete individual breakpoints using the BEFORE, AFTER, RUNTO, and DELETE primary commands.

Setting Keeps on MQ-Specific Data

In Xpediter/CICS, the KEEP command is used to continuously view a data field on the Source Listing screen (2.L). You can display various data items, such as working storage items, DFHCOMM fields, and MQ fields. You can add any MQ specific data items, such as a field within the MQMD structure, to the keep window, as long as it is defined to your program. Once displayed, you can modify data by replacing the contents of the field. This is an excellent way to test IF logic after an MQ call.

Interpreting MQ Completion and Reason Codes

Programmers commonly complain about cryptic return codes. Xpediter/CICS has MQ support within its Help facility. Enter HELP MQRC to access a convenient list of MQ completion code and reason codes.

Chapter 16. Accessing CICS Storage

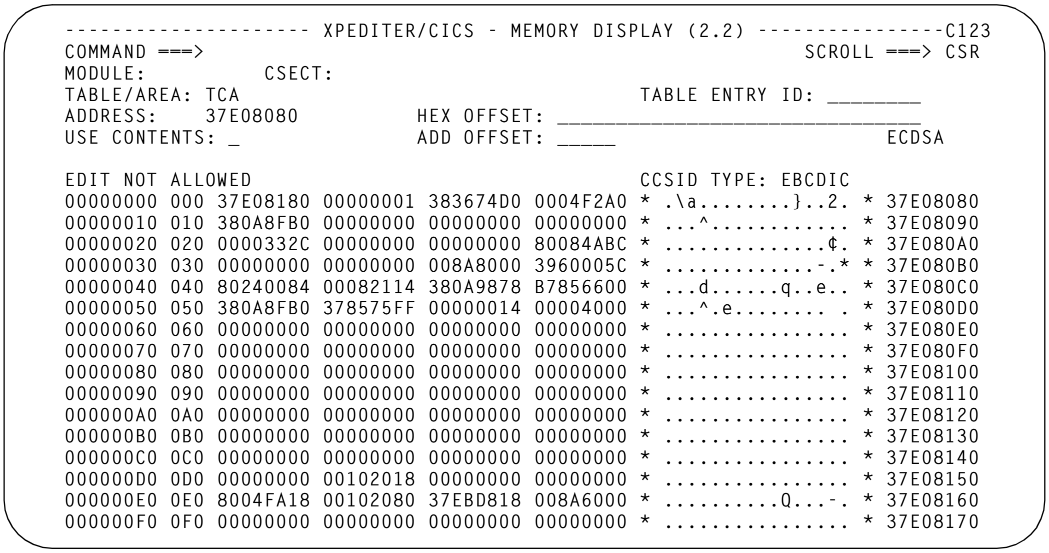
This chapter discusses ways to access and update CICS storage, including displaying CICS storage areas, DSECTs, and table entries. It also discusses how to chain through CICS storage areas and review the Select Address list.

Note: The demonstrations in this chapter should be performed in Xpediter's standard operating mode — ***not*** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

Accessing CICS Storage Areas

1. Type **XPED 2.2** on a blank CICS screen.
2. Press Enter to display the Memory Display screen (2.2) (Figure 16-1).

Figure 16-1. Displaying TCA on the Memory Display Screen (2.2)

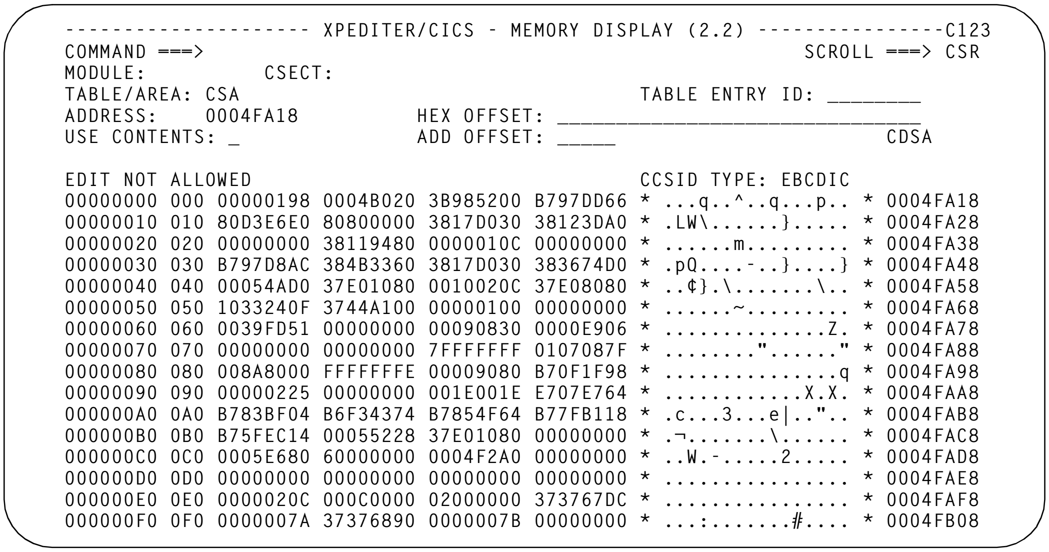


```
--------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ----------------C123
COMMAND ===>                                                     SCROLL ===> CSR
MODULE:          CSECT:
TABLE/AREA: TCA                                   TABLE ENTRY ID: ________
ADDRESS:    37E08080          HEX OFFSET: _______________________________
USE CONTENTS: _               ADD OFFSET: _____                          ECDSA

EDIT NOT ALLOWED                                  CCSID TYPE: EBCDIC
00000000 000 37E08180 00000001 383674D0 0004F2A0 * .\a........}..2. * 37E08080
00000010 010 380A8FB0 00000000 00000000 00000000 * ...^............ * 37E08090
00000020 020 0000332C 00000000 00000000 80084ABC * ..............¢. * 37E080A0
00000030 030 00000000 00000000 008A8000 3960005C * ..............-.* * 37E080B0
00000040 040 80240084 00082114 380A9878 B7856600 * ...d......q..e.. * 37E080C0
00000050 050 380A8FB0 378575FF 00000014 00004000 * ...^.e........ . * 37E080D0
00000060 060 00000000 00000000 00000000 00000000 * ................ * 37E080E0
00000070 070 00000000 00000000 00000000 00000000 * ................ * 37E080F0
00000080 080 00000000 00000000 00000000 00000000 * ................ * 37E08100
00000090 090 00000000 00000000 00000000 00000000 * ................ * 37E08110
000000A0 0A0 00000000 00000000 00000000 00000000 * ................ * 37E08120
000000B0 0B0 00000000 00000000 00000000 00000000 * ................ * 37E08130
000000C0 0C0 00000000 00000000 00000000 00000000 * ................ * 37E08140
000000D0 0D0 00000000 00102018 00000000 00000000 * ................ * 37E08150
000000E0 0E0 8004FA18 00102080 37EBD818 008A6000 * ..........Q...-. * 37E08160
000000F0 0F0 00000000 00000000 00000000 00000000 * ................ * 37E08170
```

The Memory Display screen (2.2) displays CICS storage areas. The task control area (TCA) is displayed as the default area.

3. Type **CSA** in the TABLE/AREA field to display the common system area (CSA).
4. Press Enter to display the CSA storage area (Figure 16-2).

Figure 16-2. Displaying the CSA on the Memory Display Screen (2.2)



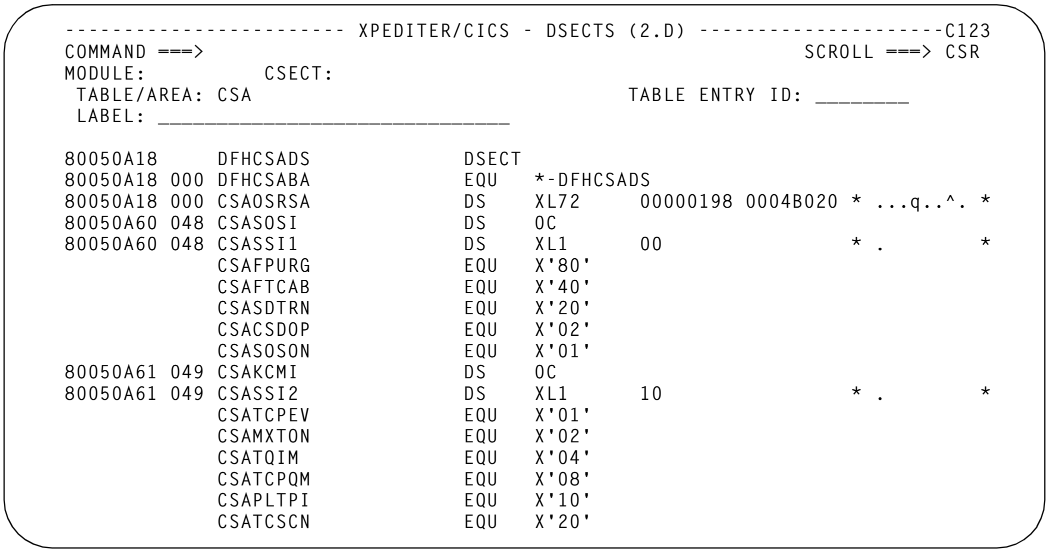
```
-------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
TABLE/AREA: CSA                                   TABLE ENTRY ID: ________
ADDRESS:     0004FA18          HEX OFFSET: _______________________________
USE CONTENTS: _                ADD OFFSET: _____                      CDSA

EDIT NOT ALLOWED                                  CCSID TYPE: EBCDIC
00000000 000 00000198 0004B020 3B985200 B797DD66 * ...q..^..q...p.. * 0004FA18
00000010 010 80D3E6E0 80800000 3817D030 38123DA0 * .LW\......}..... * 0004FA28
00000020 020 00000000 38119480 0000010C 00000000 * ......m......... * 0004FA38
00000030 030 B797D8AC 384B3360 3817D030 383674D0 * .pQ....-..}....} * 0004FA48
00000040 040 00054AD0 37E01080 0010020C 37E08080 * ..¢}.\.......\.. * 0004FA58
00000050 050 1033240F 3744A100 00000100 00000000 * ......~......... * 0004FA68
00000060 060 0039FD51 00000000 00090830 0000E906 * ..............Z. * 0004FA78
00000070 070 00000000 00000000 7FFFFFFF 0107087F * ........"......" * 0004FA88
00000080 080 008A8000 FFFFFFFE 00009080 B70F1F98 * ...............q * 0004FA98
00000090 090 00000225 00000000 001E001E E707E764 * ............X.X. * 0004FAA8
000000A0 0A0 B783BF04 B6F34374 B7854F64 B77FB118 * .c...3...e|..".. * 0004FAB8
000000B0 0B0 B75FEC14 00055228 37E01080 00000000 * .¬.......\...... * 0004FAC8
000000C0 0C0 0005E680 60000000 0004F2A0 00000000 * ..W.-.....2..... * 0004FAD8
000000D0 0D0 00000000 00000000 00000000 00000000 * ................ * 0004FAE8
000000E0 0E0 0000020C 000C0000 02000000 373767DC * ................ * 0004FAF8
000000F0 0F0 0000007A 37376890 0000007B 00000000 * ...:.......#.... * 0004FB08
```

Displaying CICS DSECTs

You may display any CICS storage area in symbolic format.

1. Type **=2.D** in the COMMAND field to display the CSA DSECT.
2. Press Enter to display the DSECTs screen (2.D) (Figure 16-3). The following options are available:
   - Access specific fields in the storage area by typing the field name in the LABEL field.
   - Browse the storage area by pressing PF7 and PF8 to scroll up and down.
3. Press PF8 to scroll down. The DSECTs screen (2.D) is scrolled to show the next page of DSECT entries.

Figure 16-3. Accessing the CSA on the DSECTs Screen (2.D)



```
----------------------- XPEDITER/CICS - DSECTS (2.D) --------------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
 TABLE/AREA: CSA                                 TABLE ENTRY ID: ________
 LABEL: ______________________________

80050A18     DFHCSADS            DSECT
80050A18 000 DFHCSABA            EQU   *-DFHCSADS
80050A18 000 CSAOSRSA            DS    XL72     00000198 0004B020 * ...q..^. *
80050A60 048 CSASOSI             DS    0C
80050A60 048 CSASSI1             DS    XL1      00                * .        *
             CSAFPURG            EQU   X'80'
             CSAFTCAB            EQU   X'40'
             CSASDTRN            EQU   X'20'
             CSACSDOP            EQU   X'02'
             CSASOSON            EQU   X'01'
80050A61 049 CSAKCMI             DS    0C
80050A61 049 CSASSI2             DS    XL1      10                * .        *
             CSATCPEV            EQU   X'01'
             CSAMXTON            EQU   X'02'
             CSATQIM             EQU   X'04'
             CSATCPQM            EQU   X'08'
             CSAPLTPI            EQU   X'10'
             CSATCSCN            EQU   X'20'
```

Chaining through CICS Storage Areas

There may be times when you have the need to chain through storage to track down an abend. Xpediter/CICS provides several methods to do this online. Four methods are described separately in this section, each of which uses the same example Memory Display screen (2.2) shown in Figure 16-4:

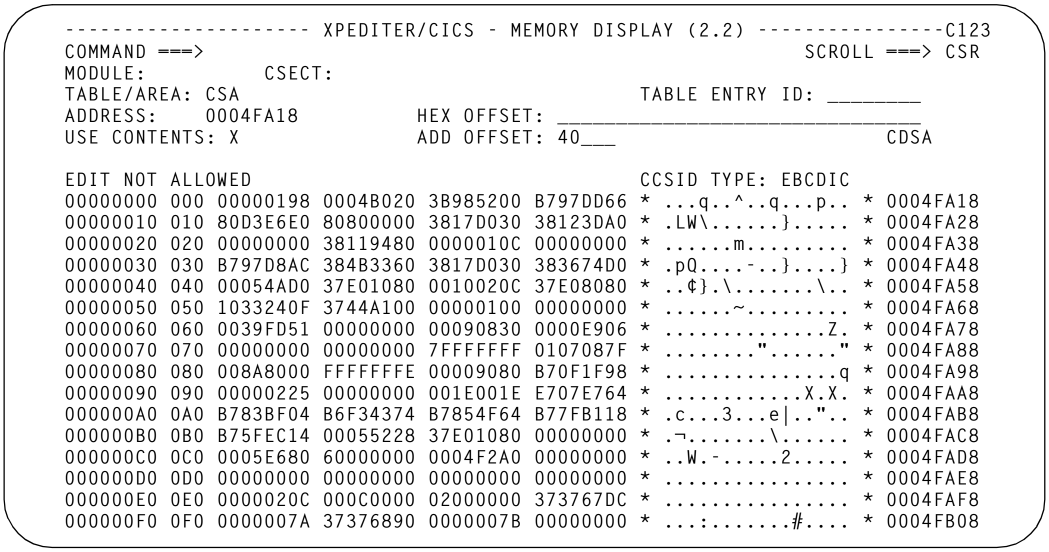
- **Method 1** — USE CONTENTS and ADD OFFSET fields
- **Method 2** — USE CONTENTS field
- **Method 3** — PF15
- **Method 4** — CHAIN command
- **Method 5** — Task Storage screen (2.S).

Method 1

Use the USE CONTENTS and ADD OFFSET fields of the Memory Display screen (2.2) to specify an address area to be displayed.

1. Transfer to the Memory Display screen (2.2) by pressing PF14 (MEMORY).

Figure 16-4. Chaining Through the CSA on the Memory Display Screen (2.2)



```
--------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) -----------------C123
COMMAND ===>                                                    SCROLL ===> CSR
MODULE:          CSECT:
TABLE/AREA: CSA                                  TABLE ENTRY ID: ________
ADDRESS:    0004FA18          HEX OFFSET: ________________________________
USE CONTENTS: X               ADD OFFSET: 40___                      CDSA

EDIT NOT ALLOWED                                 CCSID TYPE: EBCDIC
00000000 000 00000198 0004B020 3B985200 B797DD66 * ...q..^..q...p.. * 0004FA18
00000010 010 80D3E6E0 80800000 3817D030 38123DA0 * .LW\......}..... * 0004FA28
00000020 020 00000000 38119480 0000010C 00000000 * ......m......... * 0004FA38
00000030 030 B797D8AC 384B3360 3817D030 383674D0 * .pQ....-..}....} * 0004FA48
00000040 040 00054AD0 37E01080 0010020C 37E08080 * ..¢}.\.......\.. * 0004FA58
00000050 050 1033240F 3744A100 00000100 00000000 * ......~......... * 0004FA68
00000060 060 0039FD51 00000000 00090830 0000E906 * ..............Z. * 0004FA78
00000070 070 00000000 00000000 7FFFFFFF 0107087F * ........"......" * 0004FA88
00000080 080 008A8000 FFFFFFFE 00009080 B70F1F98 * ...............q * 0004FA98
00000090 090 00000225 00000000 001E001E E707E764 * ............X.X. * 0004FAA8
000000A0 0A0 B783BF04 B6F34374 B7854F64 B77FB118 * .c...3...e|..".. * 0004FAB8
000000B0 0B0 B75FEC14 00055228 37E01080 00000000 * .¬.......\...... * 0004FAC8
000000C0 0C0 0005E680 60000000 0004F2A0 00000000 * ..W.-.....2..... * 0004FAD8
000000D0 0D0 00000000 00000000 00000000 00000000 * ................ * 0004FAE8
000000E0 0E0 0000020C 000C0000 02000000 373767DC * ................ * 0004FAF8
000000F0 0F0 0000007A 37376890 0000007B 00000000 * ...:.......#.... * 0004FB08
```

2. Type an **X** in the USE CONTENTS field and **40** in the ADD OFFSET field as shown in Figure 16-4. The USE CONTENTS field specifies that Xpediter/CICS should use an address from the display area to point to another area for display. The default uses the address at offset hexadecimal 0 (meaning that there is no offset at all and that Xpediter/CICS must use the displayed address).

   The ADD OFFSET field is used to locate the address specified in the USE CONTENTS field within the display. This specifies that Xpediter/CICS should use the address found at offset hexadecimal 40 to locate another area, then display that area.

3. Press Enter. The storage area display is positioned at the address shown at an offset of hexadecimal 40 from the start of the CSA.

   Notice that the ADDRESS field has changed to show the new address.

Method 2

Use the USE CONTENTS field without the ADD OFFSET field to have Xpediter/CICS use the first four bytes of the display as an address.

1. Type **X** in the USE CONTENTS field.
2. Press Enter. The Memory Display screen (2.2) displays the storage area at the address located at hexadecimal offset 0.

Method 3

Use PF15 to select addresses.

1. Type **CSA** in the TABLE/AREA field and press Enter.
2. Move the cursor to the address at an offset of hexadecimal 04C on the screen.
3. Press PF15. The Memory Display screen (2.2) displays the storage area located at the address at a hexadecimal offset of 04C.

Method 4

Use the CHAIN command to chain to the first address displayed in the storage area of the screen. The CHAIN command works like the USE CONTENTS field described in "Method 1" on page 16-3 and "Method 2".

Because CHAIN is a primary command, it can be assigned to a PF key. This makes chaining a one-key function. CHAIN can also be used with an offset. For example: CHAIN 4 or CHAIN 2C.

1. Type **CHAIN** in the COMMAND field.
2. Press Enter. The Memory Display screen (2.2) displays the storage area at the address located at hexadecimal offset 0.

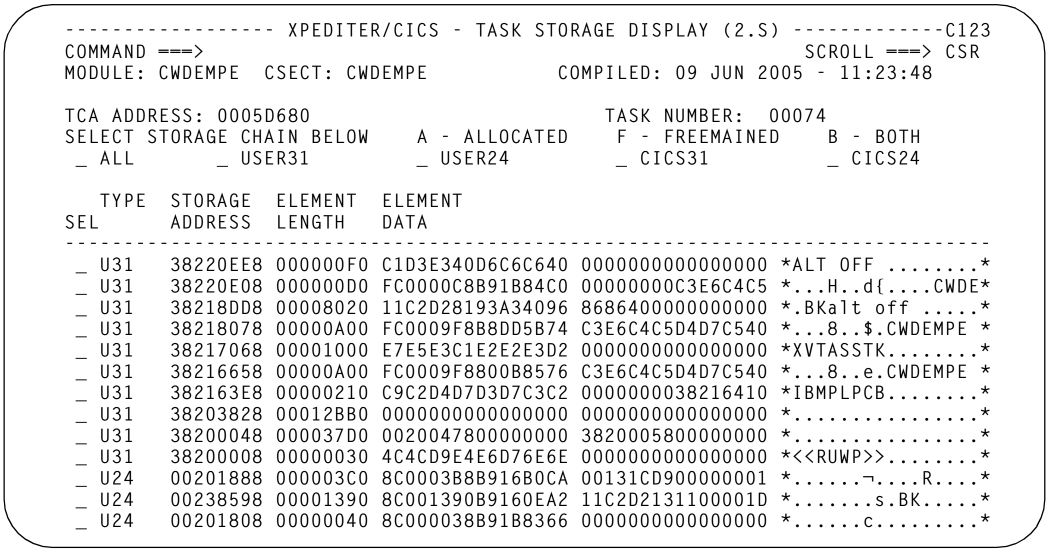
Method 5

Since CICS control block changes have made it increasingly difficult to chain through a task's storage areas, Xpediter/CICS has provided a Task Storage screen (2.S) shown in Figure 16-5 on page 16-5 that allows you to easily display the areas on a task's USER31, USER24, CICS31, and CICS24 DSA chains. You can request any or all of these chains and allocated and/or freemained areas. You can then select a specific area by entering an **S** in the SEL column next to the desired area. Xpediter will transfer to the Memory Display screen (2.2 or 9.2, based on Xpediter session type). The END command (default PF3) will return you to the list of storage areas on the Task Storage screen with the last selected area positioned on the top line.

1. Access the Task Storage screen by entering the letter **S** on the Debugging Facilities Menu (2) or by entering **=2.S** in the COMMAND field. You can also access this screen as described in the section entitled "Transferring Between Screens" in Chapter 2 of the *Xpediter/CICS Reference Manual*.

   The Task Storage screen (Figure 16-5 on page 16-5) will be displayed with storage for the current task (the default) showing all allocated areas on all four of the DSA chains. You can vary the task selected and/or the combination of allocated and/or freemained areas on the four DSA chains. Only freed areas still on the chains are available—***not*** all areas freed anytime during the duration of the transaction. It should also be noted that areas on the freemained chains may occasionally be allocated by CICS while Xpediter is processing your viewing request.

Figure 16-5. Task Storage Screen (2.S)



```
------------------ XPEDITER/CICS - TASK STORAGE DISPLAY (2.S) -------------C123
COMMAND ===>                                                     SCROLL ===> CSR
MODULE: CWDEMPE  CSECT: CWDEMPE          COMPILED: 09 JUN 2005 - 11:23:48

TCA ADDRESS: 0005D680                        TASK NUMBER:  00074
SELECT STORAGE CHAIN BELOW    A - ALLOCATED    F - FREEMAINED    B - BOTH
 _ ALL       _ USER31         _ USER24         _ CICS31          _ CICS24

   TYPE  STORAGE  ELEMENT  ELEMENT
SEL      ADDRESS  LENGTH   DATA
--------------------------------------------------------------------------------
 _ U31   38220EE8 000000F0 C1D3E340D6C6C640 0000000000000000 *ALT OFF ........*
 _ U31   38220E08 000000D0 FC0000C8B91B84C0 00000000C3E6C4C5 *...H..d{....CWDE*
 _ U31   38218DD8 00008020 11C2D28193A34096 8686400000000000 *.BKalt off .....*
 _ U31   38218078 00000A00 FC0009F8B8DD5B74 C3E6C4C5D4D7C540 *...8..$.CWDEMPE *
 _ U31   38217068 00001000 E7E5E3C1E2E2E3D2 0000000000000000 *XVTASSTK........*
 _ U31   38216658 00000A00 FC0009F8800B8576 C3E6C4C5D4D7C540 *...8..e.CWDEMPE *
 _ U31   382163E8 00000210 C9C2D4D7D3D7C3C2 0000000038216410 *IBMPLPCB........*
 _ U31   38203828 00012BB0 0000000000000000 0000000000000000 *................*
 _ U31   38200048 000037D0 0020047800000000 3820005800000000 *................*
 _ U31   38200008 00000030 4C4CD9E4E6D76E6E 0000000000000000 *<<RUWP>>........*
 _ U24   00201888 000003C0 8C0003B8B916B0CA 00131CD900000001 *......¬....R....*
 _ U24   00238598 00001390 8C001390B9160EA2 11C2D2131100001D *.......s.BK.....*
 _ U24   00201808 00000040 8C000038B91B8366 0000000000000000 *......c.........*
```

2. Select the area you want to view by typing an **S** in the SEL column of the desired area and pressing Enter. The Memory Display screen (2.2 or 9.2, based on Xpediter session type) will be displayed. The first 16 bytes of the areas are displayed in hex and character mode to make selection easier.
3. After viewing and/or updating an area, use the END command (default PF3) to return to the Task Storage screen. The last area selected will be positioned on the top line.
4. You can page through the remaining areas, selecting and viewing any of interest. Only one area can be selected at a time.

   You can also use the LOCATE command with a hex address to determine whether an address is contained in any of the areas on the selected chains. If it is, that area will be positioned at the top of the screen, and you can select it to view the entire area.

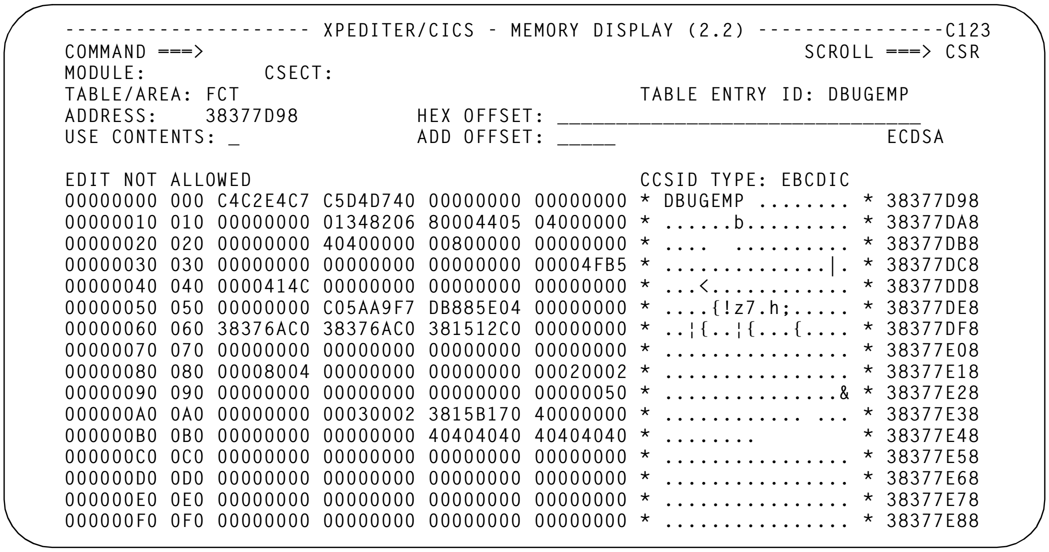
Displaying CICS Table Entries

In this example, you will display several CICS table areas by using the TABLE ENTRY ID field of the Storage Areas screen to access a specific table entry directly.

1. Type **FCT** (for file control table) in the TABLE/AREA field of the Memory Display screen (2.2) (Figure 16-6).
2. Type **DBUGEMP** in the TABLE ENTRY ID field. DBUGEMP is a file used in the Xpediter/CICS demonstration programs.
3. Press Enter to display the FCT entry for file DBUGEMP.

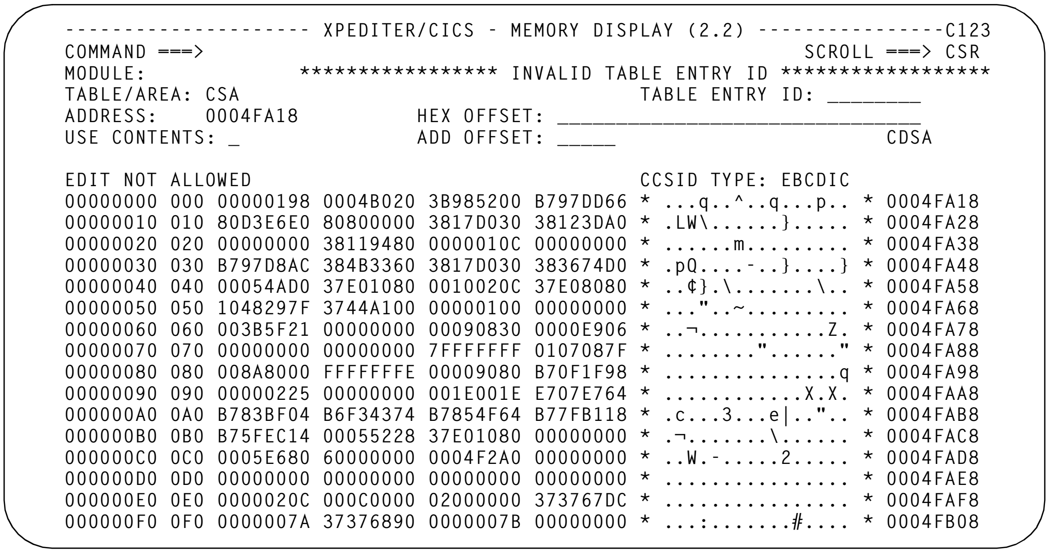
   Note: If your file resource definition does not exist, after you type FCT in the TABLE/AREA field, you see the message shown in Figure 16-7 on page 16-6.

Figure 16-6. Displaying the FCT on the Memory Display Screen (2.2)



```
-------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
TABLE/AREA: FCT                                  TABLE ENTRY ID: DBUGEMP
ADDRESS:    38377D98          HEX OFFSET: _______________________________
USE CONTENTS: _               ADD OFFSET: _____                      ECDSA

EDIT NOT ALLOWED                                 CCSID TYPE: EBCDIC
00000000 000 C4C2E4C7 C5D4D740 00000000 00000000 * DBUGEMP ........ * 38377D98
00000010 010 00000000 01348206 80004405 04000000 * ......b......... * 38377DA8
00000020 020 00000000 40400000 00800000 00000000 * ....  .......... * 38377DB8
00000030 030 00000000 00000000 00000000 00004FB5 * ..............|. * 38377DC8
00000040 040 0000414C 00000000 00000000 00000000 * ...<............ * 38377DD8
00000050 050 00000000 C05AA9F7 DB885E04 00000000 * ....{!z7.h;..... * 38377DE8
00000060 060 38376AC0 38376AC0 381512C0 00000000 * ..¦{..¦{...{.... * 38377DF8
00000070 070 00000000 00000000 00000000 00000000 * ................ * 38377E08
00000080 080 00008004 00000000 00000000 00020002 * ................ * 38377E18
00000090 090 00000000 00000000 00000000 00000050 * ...............& * 38377E28
000000A0 0A0 00000000 00030002 3815B170 40000000 * ............ ... * 38377E38
000000B0 0B0 00000000 00000000 40404040 40404040 * ........         * 38377E48
000000C0 0C0 00000000 00000000 00000000 00000000 * ................ * 38377E58
000000D0 0D0 00000000 00000000 00000000 00000000 * ................ * 38377E68
000000E0 0E0 00000000 00000000 00000000 00000000 * ................ * 38377E78
000000F0 0F0 00000000 00000000 00000000 00000000 * ................ * 38377E88
```

Figure 16-7. Invalid Table Entry ID Message



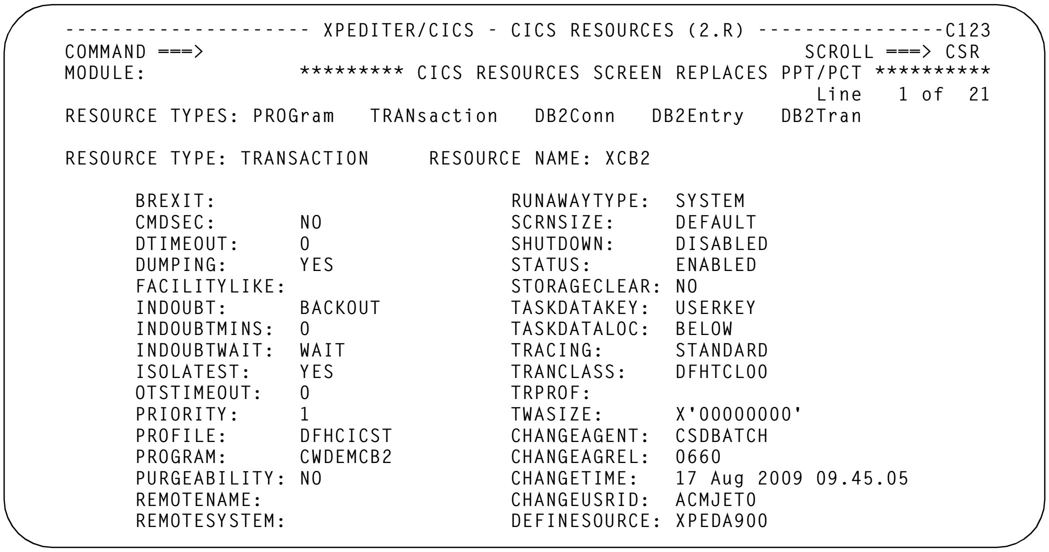
```
-------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:             ***************** INVALID TABLE ENTRY ID *****************
TABLE/AREA: CSA                                  TABLE ENTRY ID: ________
ADDRESS:    0004FA18          HEX OFFSET: _______________________________
USE CONTENTS: _               ADD OFFSET: _____                      CDSA

EDIT NOT ALLOWED                                 CCSID TYPE: EBCDIC
00000000 000 00000198 0004B020 3B985200 B797DD66 * ...q..^..q...p.. * 0004FA18
00000010 010 80D3E6E0 80800000 3817D030 38123DA0 * .LW\......}..... * 0004FA28
00000020 020 00000000 38119480 0000010C 00000000 * ......m......... * 0004FA38
00000030 030 B797D8AC 384B3360 3817D030 383674D0 * .pQ....-..}....} * 0004FA48
00000040 040 00054AD0 37E01080 0010020C 37E08080 * ..¢}.\.......\.. * 0004FA58
00000050 050 1048297F 3744A100 00000100 00000000 * ..."..~......... * 0004FA68
00000060 060 003B5F21 00000000 00090830 0000E906 * ..¬...........Z. * 0004FA78
00000070 070 00000000 00000000 7FFFFFFF 0107087F * ........"......" * 0004FA88
00000080 080 008A8000 FFFFFFFE 00009080 B70F1F98 * ...............q * 0004FA98
00000090 090 00000225 00000000 001E001E E707E764 * ............X.X. * 0004FAA8
000000A0 0A0 B783BF04 B6F34374 B7854F64 B77FB118 * .c...3...e|..".. * 0004FAB8
000000B0 0B0 B75FEC14 00055228 37E01080 00000000 * .¬.......\...... * 0004FAC8
000000C0 0C0 0005E680 60000000 0004F2A0 00000000 * ..W.-.....2..... * 0004FAD8
000000D0 0D0 00000000 00000000 00000000 00000000 * ................ * 0004FAE8
000000E0 0E0 0000020C 000C0000 02000000 373767DC * ................ * 0004FAF8
000000F0 0F0 0000007A 37376890 0000007B 00000000 * ...:.......#.... * 0004FB08
```

Displaying Program and Transaction Resources

CICS no longer supports the use of the PCT and PPT table entry keywords. You may access the CICS Resources screen (2.R) to view the attributes of a program or transaction.

If you do request a PCT entry for XPLE, control automatically passes to the CICS Resources screen (2.R) shown in Figure 16-8. From the CICS Resources screen (2.R), you may view the formatted resource information for your program or transaction.

Figure 16-8. CICS Resources Screen (2.R)



```
-------------------- XPEDITER/CICS - CICS RESOURCES (2.R) ---------------C123
COMMAND ===>                                                SCROLL ===> CSR
MODULE:            ********* CICS RESOURCES SCREEN REPLACES PPT/PCT **********
                                                             Line   1 of  21
RESOURCE TYPES: PROGram   TRANsaction   DB2Conn   DB2Entry   DB2Tran

RESOURCE TYPE: TRANSACTION     RESOURCE NAME: XCB2

      BREXIT:                         RUNAWAYTYPE:  SYSTEM
      CMDSEC:       NO                SCRNSIZE:     DEFAULT
      DTIMEOUT:     0                 SHUTDOWN:     DISABLED
      DUMPING:      YES               STATUS:       ENABLED
      FACILITYLIKE:                   STORAGECLEAR: NO
      INDOUBT:      BACKOUT           TASKDATAKEY:  USERKEY
      INDOUBTMINS:  0                 TASKDATALOC:  BELOW
      INDOUBTWAIT:  WAIT              TRACING:      STANDARD
      ISOLATEST:    YES               TRANCLASS:    DFHTCL00
      OTSTIMEOUT:   0                 TRPROF:
      PRIORITY:     1                 TWASIZE:      X'00000000'
      PROFILE:      DFHCICST          CHANGEAGENT:  CSDBATCH
      PROGRAM:      CWDEMCB2          CHANGEAGREL:  0660
      PURGEABILITY: NO                CHANGETIME:   17 Aug 2009 09.45.05
      REMOTENAME:                     CHANGEUSRID:  ACMJETO
      REMOTESYSTEM:                   DEFINESOURCE: XPEDA900
```

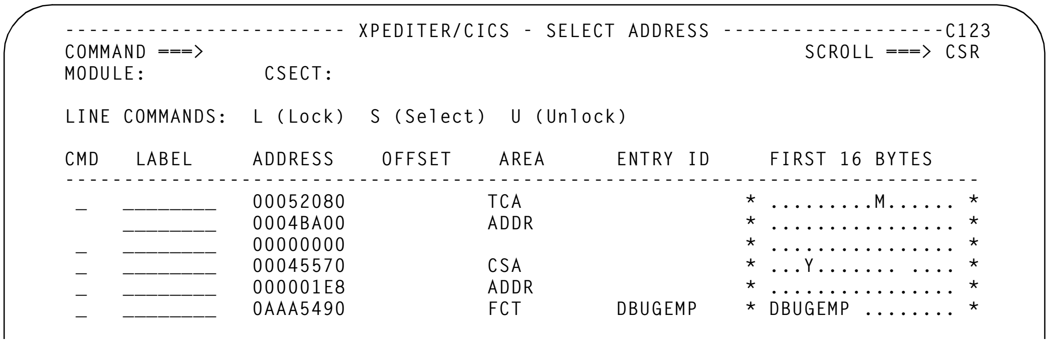
Reviewing the Select Address List

Xpediter/CICS keeps track of storage areas that were accessed during a debugging session by saving the address of the areas in a circular list.

1. To display the Select Address screen, position the cursor in any area of the Memory Display screen (2.2) except the data display and press PF15. The Select Address screen appears as shown in Figure 16-9.

   This list is used to review a debugging session without having to recreate the session. Notice that all the table areas accessed on the Memory Display screen (2.2) in the previous sections of this chapter are listed here.

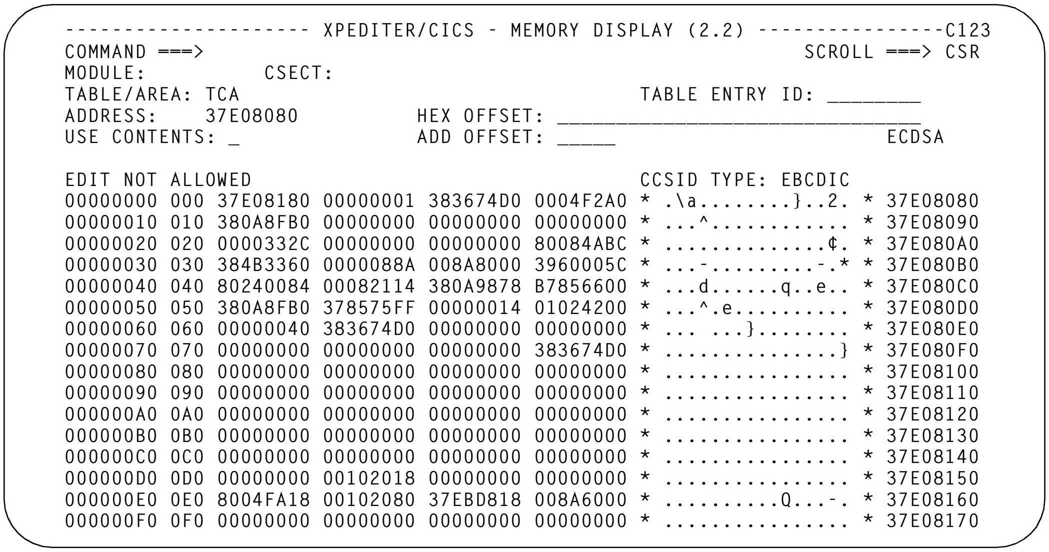
Figure 16-9. Select Address Screen



```
----------------------- XPEDITER/CICS - SELECT ADDRESS ------------------C123
COMMAND ===>                                                SCROLL ===> CSR
MODULE:           CSECT:

LINE COMMANDS:  L (Lock)  S (Select)  U (Unlock)

CMD   LABEL     ADDRESS    OFFSET    AREA      ENTRY ID     FIRST 16 BYTES
-----------------------------------------------------------------------------
 _   ________   00052080            TCA                    * .........M...... *
     ________   0004BA00            ADDR                   * ................ *
 _   ________   00000000                                   * ................ *
 _   ________   00045570            CSA                    * ...Y....... .... *
 _   ________   000001E8            ADDR                   * ................ *
 _   ________   0AAA5490            FCT       DBUGEMP      * DBUGEMP ........ *
```

2. Type **S** in the CMD column next to the desired address to review any of these screens.
3. Press Enter to display the Memory Display screen (2.2) (Figure 16-10).

Figure 16-10. Memory Display Screen (2.2) from the Select Address Screen



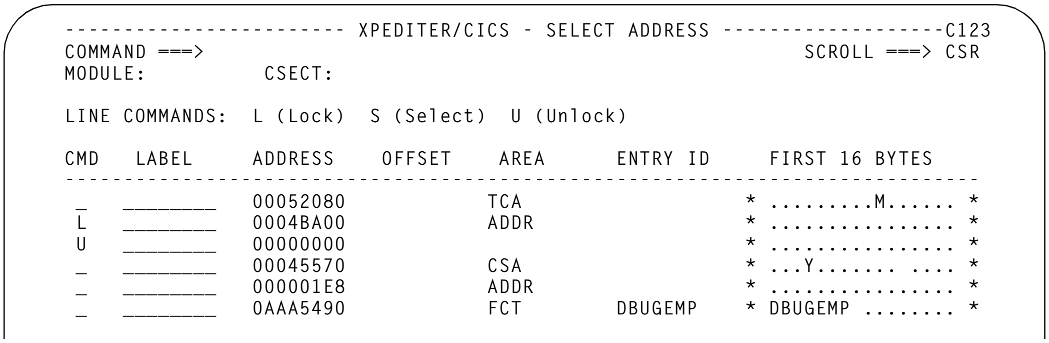
```
-------------------- XPEDITER/CICS - MEMORY DISPLAY (2.2) ---------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE:         CSECT:
TABLE/AREA: TCA                                 TABLE ENTRY ID: ________
ADDRESS:    37E08080          HEX OFFSET: _______________________________
USE CONTENTS: _               ADD OFFSET: _____                      ECDSA

EDIT NOT ALLOWED                                CCSID TYPE: EBCDIC
00000000 000 37E08180 00000001 383674D0 0004F2A0 * .\a........}..2. * 37E08080
00000010 010 380A8FB0 00000000 00000000 00000000 * ...^............ * 37E08090
00000020 020 0000332C 00000000 00000000 80084ABC * ..............¢. * 37E080A0
00000030 030 384B3360 0000088A 008A8000 3960005C * ...-.........-.* * 37E080B0
00000040 040 80240084 00082114 380A9878 B7856600 * ...d......q..e.. * 37E080C0
00000050 050 380A8FB0 378575FF 00000014 01024200 * ...^.e.......... * 37E080D0
00000060 060 00000040 383674D0 00000000 00000000 * ... ...}........ * 37E080E0
00000070 070 00000000 00000000 00000000 383674D0 * ...............} * 37E080F0
00000080 080 00000000 00000000 00000000 00000000 * ................ * 37E08100
00000090 090 00000000 00000000 00000000 00000000 * ................ * 37E08110
000000A0 0A0 00000000 00000000 00000000 00000000 * ................ * 37E08120
000000B0 0B0 00000000 00000000 00000000 00000000 * ................ * 37E08130
000000C0 0C0 00000000 00000000 00000000 00000000 * ................ * 37E08140
000000D0 0D0 00000000 00102018 00000000 00000000 * ................ * 37E08150
000000E0 0E0 8004FA18 00102080 37EBD818 008A6000 * ..........Q...-. * 37E08160
000000F0 0F0 00000000 00000000 00000000 00000000 * ................ * 37E08170
```

4. Press PF11. The next area on the select list is displayed.

   The list is especially useful for checking on possible errors during a debugging session. You can review several screens and display additional storage areas. These new areas are also saved in the list.

   Because the Select Address screen is a circular list, Xpediter/CICS removes the oldest entries when there are more than 16 addresses in it. If you need to retain a particular entry, use the L (Lock) line command to lock an entry on the list. Use the U (Unlock) line command to free these entries. Type these commands under the CMD column as shown in Figure 16-11.

Figure 16-11. Address Entries



```
----------------------- XPEDITER/CICS - SELECT ADDRESS ------------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE:         CSECT:

LINE COMMANDS:  L (Lock)  S (Select)  U (Unlock)

CMD   LABEL     ADDRESS    OFFSET    AREA      ENTRY ID     FIRST 16 BYTES
---------------------------------------------------------------------------
 _   ________   00052080            TCA                   * .........M...... *
 L   ________   0004BA00            ADDR                  * ................ *
 U   ________   00000000                                  * ................ *
 _   ________   00045570            CSA                   * ...Y....... .... *
 _   ________   000001E8            ADDR                  * ................ *
 _   ________   0AAA5490            FCT       DBUGEMP     * DBUGEMP ........ *
```

You can review the entire address list screen by using PF10 and PF11 to scroll through the list, displaying the contents of memory at each of the addresses in the list. PF10 will display the previous address in the list, and PF11 will display the next address in the list.

Remember to end the session as described in "Exiting Xpediter/CICS and Ending a Debugging Session" on page 2-6.

Chapter 17. Using Global Storage Protection

This chapter discusses the global storage protection facilities of Xpediter/CICS available through the XPSP transaction. This chapter is intended for the person responsible for maintaining the Xpediter/CICS system at the site. The chapter shows how to set region-wide and automatic storage protection, and monitor storage violations. It also discusses how to define system labels and storage exceptions.

Note: The demonstrations in this chapter should be performed in Xpediter's standard operating mode — ***not*** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

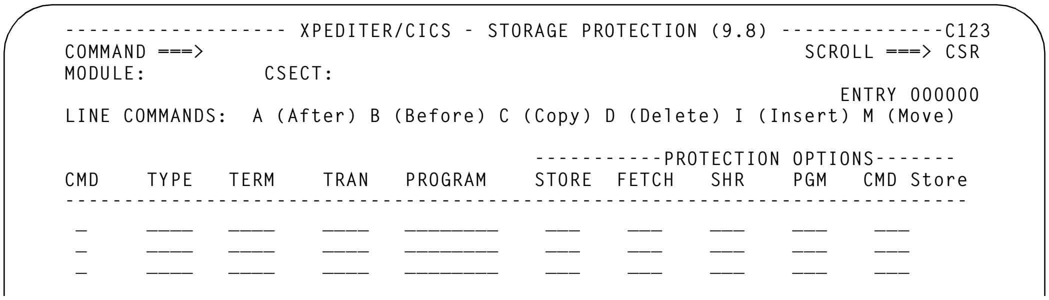
Setting Region-Wide Storage Protection

Xpediter/CICS is designed to protect CICS regions against storage violations. You can set storage protection for an entire region by individual program, transaction, terminal, or any combination of these items. You can make entries to protect storage, fetch access, shared storage, program storage, and storage by CICS command level commands — or by combining these items.

This example demonstrates how to set storage protection for a region through XPSP.

1. Type **XPSP 9.8** from a blank CICS screen and press Enter to display the Storage Protection screen (9.8) (Figure 17-1).

Figure 17-1. Making Storage Protection Entries on the Storage Protection Screen (9.8)



```
------------------ XPEDITER/CICS - STORAGE PROTECTION (9.8) --------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
                                                                  ENTRY 000000
LINE COMMANDS:  A (After) B (Before) C (Copy) D (Delete) I (Insert) M (Move)

                                        -----------PROTECTION OPTIONS-------
CMD    TYPE   TERM    TRAN   PROGRAM    STORE  FETCH   SHR    PGM   CMD Store
-----------------------------------------------------------------------------
 _     ____   ____    ____   ________    ___    ___    ___    ___    ___
 _     ____   ____    ____   ________    ___    ___    ___    ___    ___
 _     ____   ____    ____   ________    ___    ___    ___    ___    ___
```

2. Type **SYST** in the TYPE field. There are three types of storage protection entries:

   – **SYST** entries created by XPSP users or during product initialization. They always proceed other entries in the table and are always evaluated first. These entries can only be deleted by XPSP users.

   – **USER** entries follow SYST entries. They can be modified from either Storage Protection screen (1.8 or 9.8).

   – **AUTO** entries are automatically added to the table whenever automatic storage protection is set ON and any Xpediter/CICS transaction is used. See "Setting Automatic Storage Protection" on page 17-2 for more information.

3. Type an asterisk (*****) in the TERM field to specify all terminals.

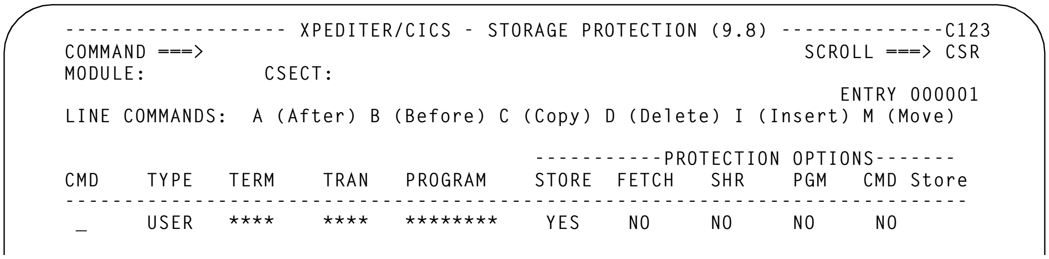
4. Type **XP**** in the TRAN field to specify any transaction that begins with the letters XP.
5. Type **CWDEMPE** in the PROGRAM field.
6. Type **Y** in the STORE field under PROTECTION OPTIONS.
7. Press Enter to redisplay the Storage Protection screen (9.8). Xpediter/CICS uses the default value of NO for FETCH, SHR, PGM, and CMD Store.

   These entries indicate that the program CWDEMPE is prevented from attempting to cause storage violations whenever it is invoked from any terminal with a transaction that starts with the letters XP.

Sample Storage Protection Entries

The settings shown in Figure 17-2 can be used to ensure that all transactions are monitored for storage violations, yet allow Xpediter/CICS users to tailor their own storage protection. Entries are searched in the order in which they appear in the table. This entry should be the last one in the table—all other users' entries precede this one.

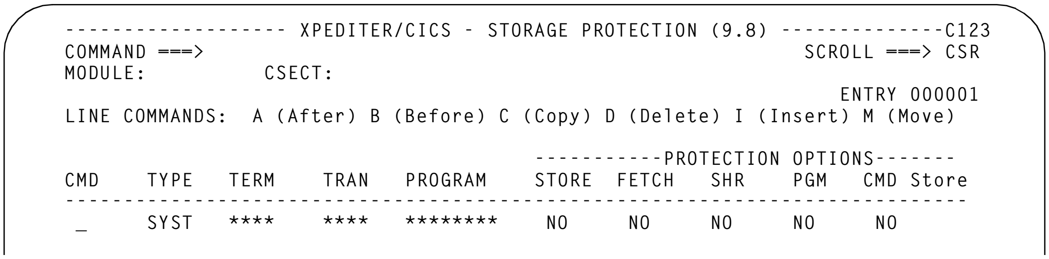
Figure 17-2. Monitoring Transactions on the Storage Protection Screen (9.8)



```
------------------ XPEDITER/CICS - STORAGE PROTECTION (9.8) -------------C123
COMMAND ===>                                               SCROLL ===> CSR
MODULE:          CSECT:
                                                              ENTRY 000001
LINE COMMANDS:  A (After) B (Before) C (Copy) D (Delete) I (Insert) M (Move)

                                        -----------PROTECTION OPTIONS-------
CMD    TYPE   TERM    TRAN   PROGRAM    STORE  FETCH   SHR    PGM   CMD Store
-----------------------------------------------------------------------------
 _     USER   ****    ****   ********    YES    NO     NO     NO     NO
```

The settings shown in Figure 17-3 turn off monitoring without destroying the current table entries. These settings must appear first in the table.

Figure 17-3. Turning Off Monitoring on the Storage Protection Screen (9.8)



```
------------------ XPEDITER/CICS - STORAGE PROTECTION (9.8) -------------C123
COMMAND ===>                                               SCROLL ===> CSR
MODULE:          CSECT:
                                                              ENTRY 000001
LINE COMMANDS:  A (After) B (Before) C (Copy) D (Delete) I (Insert) M (Move)

                                        -----------PROTECTION OPTIONS-------
CMD    TYPE   TERM    TRAN   PROGRAM    STORE  FETCH   SHR    PGM   CMD Store
-----------------------------------------------------------------------------
 _     SYST   ****    ****   ********    NO     NO     NO     NO     NO
```

Setting Automatic Storage Protection

Xpediter/CICS is shipped with the storage protection function turned OFF for all three transactions (XPED, XPRT, and XPSP). Although these settings prevent Xpediter/CICS from intercepting storage violations, they may be acceptable for several reasons. First, if applications are run in test regions, the impact of storage violations is probably not as great as in a production environment. Second, storage monitoring requires additional system resources to inspect each instruction before it executes. Thus, in most cases, the need for protection is less than the cost of providing protection. In these cases, storage protection should be turned OFF.

Note: If Xpediter/CICS is operating in Utilities Mode or Diagnosis/Utilities Mode, the storage protection function is ***not*** available, regardless of global table parameter settings or attempted user overrides. These modes are designed for use in throughput-critical CICS regions.

Activating Test Region Storage Protection

There are times when you can't afford to expose your test region to outages caused by storage violations. This is the time to change the default so that storage protection is automatically turned ON whenever a test session is started.

The global parameter DEFPROT (default OFF,OFF,OFF) controls automatic storage protection for all three transactions (XPED, XPRT, and XPSP). When DEFPROT is set to ON for any Xpediter/CICS transaction, storage protection is automatically provided whenever a user enters that transaction. For example, setting DEFPROT to (ON,OFF,OFF) automatically sets storage protection whenever the XPED transaction is used.

The DEFPROT settings can be overridden by individual users, allowing them to set up different protection for different testing situations. The SET PROTECT command turns storage protection ON or OFF for the STORE protection option. The PROTECT field on the Set Profile Defaults screen (0.1) indicates the storage protection setting for an individual session.

Allowing Storage Violations

Xpediter/CICS storage protection rules are very strict. Only storage that belongs to the program can be updated. Anything else is intercepted as a violation. There are times when an application programmer may disagree with the Xpediter/CICS assessment of what is a violation. With the ALLOW command, you can allow the trapped violation to occur, yet still provide protection for the rest of the test.

The ALLOWCM global parameter controls access to the ALLOW command when using Xpediter/CICS transactions. Xpediter/CICS is shipped with ALLOW set to OFF (no) for the XPED/XPRT transactions and ON (yes) for the XPSP transaction. Xpediter's three-transaction design gives you the ability to control who has access to allow storage violations by turning ALLOWCM ON for one transaction and OFF for the others.

Note: For more information about the global parameters, refer to the *Xpediter/CICS Installation Guide*.

Defining System Labels

Defining storage exceptions is another way to allow non-destructive storage violations to occur. Many sites have in-house or vendor-supplied programs that break the rules for storage protection. To avoid having storage violations reported for this activity, Xpediter/CICS provides two screens to define, label, and exempt certain areas from storage protection. Xpediter/CICS can then monitor a program for storage violations, yet define areas that are exempt from storage violation protection. With this facility, your program can modify areas without Xpediter/CICS considering the modification to be a storage violation.

The Define System Labels screen (9.9) is used to define areas to be exempted from storage protection.

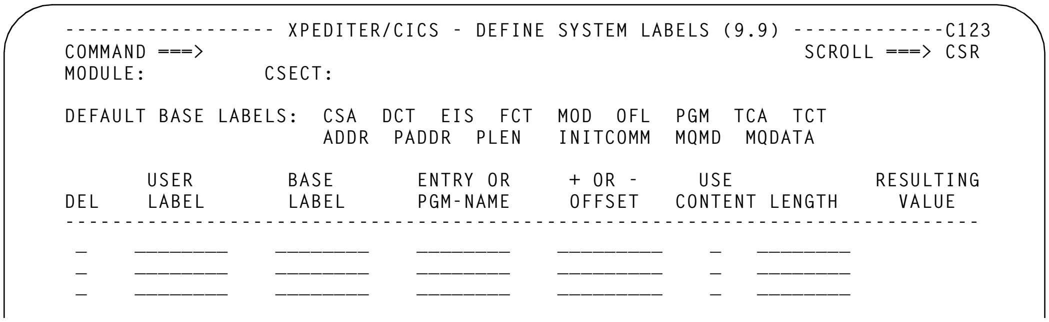
1. Type **=9.9** in the COMMAND field of any screen and press Enter to display the Define System Labels screen (9.9) (Figure 17-4).

In this example, assume that program CWDEMPE needs to modify the TRAN ID field of the execute interface block (EIB). Identify and label the field as EIBTRNID to exempt it from storage protection.

The TRAN ID field is located in the EIB. Two entries are made: one to locate the start of the EIB (EISEIBAD), the second to locate the TRAN ID field within the EIB (EIBTRNID).

Note: Program CWDEMPE is used here only as an example. It does not actually update the EIB.

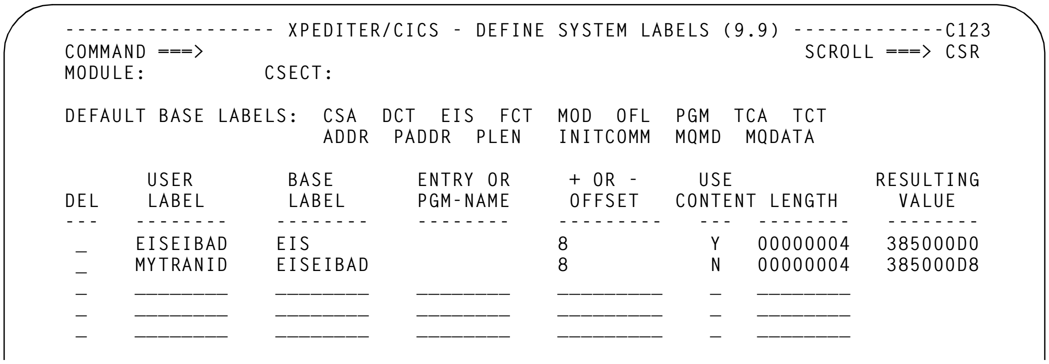
Figure 17-4. Define System Labels Screen (9.9)



```
----------------- XPEDITER/CICS - DEFINE SYSTEM LABELS (9.9) -------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE:          CSECT:

DEFAULT BASE LABELS:  CSA   DCT   EIS   FCT   MOD   OFL   PGM   TCA   TCT
                      ADDR  PADDR  PLEN   INITCOMM  MQMD  MQDATA

       USER        BASE        ENTRY OR     + OR -     USE            RESULTING
DEL    LABEL       LABEL       PGM-NAME     OFFSET   CONTENT LENGTH     VALUE
-------------------------------------------------------------------------------
 _    ________    ________    ________    _________     _    ________
 _    ________    ________    ________    _________     _    ________
 _    ________    ________    ________    _________     _    ________
```

2. To locate the starting address of the EIB, type **EISEIBAD** in the USER LABEL field. This defines a label for the beginning of the EIB.
3. Type **EIS** in the BASE LABEL field.
4. Type **8** in the + OR - OFFSET field. This value is the offset from the beginning of the EIS where the EIB address is located.
5. Type **Y** in the USE CONTENT field to specify that the data located eight bytes into the EIS is to be used as an address.
6. Type **4** in the LENGTH field to indicate the length of the area to be used.
7. Press Enter. The first entry is recorded.
8. Type **MYTRANID** in the USER LABEL field. This identifies the EIBTRNID.
9. Type **EISEIBAD** in the BASE LABEL field. This is the label defined in step 2.
10. Type **8** in the + OR - OFFSET field.
11. Type **N** in the USE CONTENT field to indicate that the data found at this address is *not* to be used as an address.
12. Type **4** in the LENGTH field.
13. Press Enter. If the addresses can be resolved, the RESULTING VALUE field is updated as shown in Figure 17-5. All addresses and lengths are resolved at the time they are used.

Figure 17-5. EIB Entries on Define System Labels Screen (9.9)



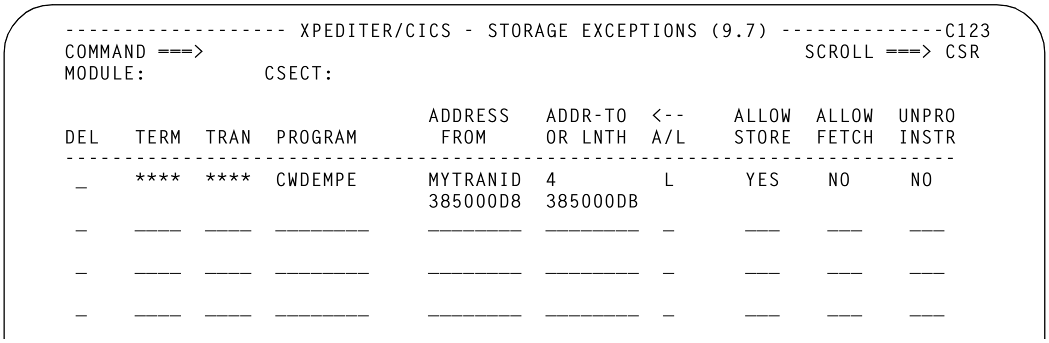
```
----------------- XPEDITER/CICS - DEFINE SYSTEM LABELS (9.9) -------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:

DEFAULT BASE LABELS:  CSA  DCT  EIS  FCT  MOD  OFL  PGM  TCA  TCT
                      ADDR  PADDR  PLEN   INITCOMM  MQMD  MQDATA

       USER        BASE       ENTRY OR     + OR -     USE              RESULTING
DEL    LABEL       LABEL      PGM-NAME     OFFSET   CONTENT LENGTH       VALUE
---   --------    --------    --------    ---------   ---   --------   --------
 _    EISEIBAD    EIS                     8            Y    00000004   385000D0
 _    MYTRANID    EISEIBAD                8            N    00000004   385000D8
 _    ________    ________    ________    _________    _    ________
 _    ________    ________    ________    _________    _    ________
 _    ________    ________    ________    _________    _    ________
```

Defining Storage Exceptions

Now that you have identified and labeled the MYTRANID field, you can make an entry to exempt this area from storage protection. The next entry allows program CWDEMPE to update this field.

1. Type **=9.7** in the COMMAND field and press Enter to access the Storage Exceptions screen. The following steps will create the entry seen in Figure 17-6.

Figure 17-6. Exceptions Entered on the Storage Exceptions Screen (9.7)



```
------------------ XPEDITER/CICS - STORAGE EXCEPTIONS (9.7) --------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:

                                ADDRESS   ADDR-TO  <--    ALLOW  ALLOW  UNPRO
DEL   TERM  TRAN  PROGRAM        FROM     OR LNTH  A/L    STORE  FETCH  INSTR
-------------------------------------------------------------------------------
 _    ****  ****  CWDEMPE       MYTRANID  4         L      YES    NO     NO
                                385000D8  385000DB
 _    ____  ____  ________      ________  ________  _      ___    ___    ___

 _    ____  ____  ________      ________  ________  _      ___    ___    ___

 _    ____  ____  ________      ________  ________  _      ___    ___    ___
```

2. Type an asterisk (*) in the TERM field to specify all terminals.
3. Type an asterisk (*) in the TRAN field to specify all transactions.
4. Type **CWDEMPE** in the PROGRAM field.
5. Type **MYTRANID** in the ADDRESS FROM field.
6. Type **4** in the ADDR-TO OR LNTH field.
7. Type **L** (for length) in the A/L field.
8. Type **Y** in the ALLOW STORE field.
9. Press Enter. The default value NO is taken for ALLOW FETCH and UNPRO INSTR.

These entries specify that whenever CWDEMPE is executed from any terminal or transaction, it can modify the four-byte area beginning at the label EIBTRNID.

When evaluated with storage protection active, Xpediter/CICS monitors CWDEMPE for storage violations but allows updates to the TRAN ID field.

Note: Entries made on the Storage Exceptions screen (9.7), the Storage Protection screen (9.8), and the Define System Labels screen (9.9) stay in effect until they are deleted or Xpediter is turned off.

Remember to end your session as described in "Exiting Xpediter/CICS and Ending a Debugging Session" on page 2-6.

Monitoring Storage Violations

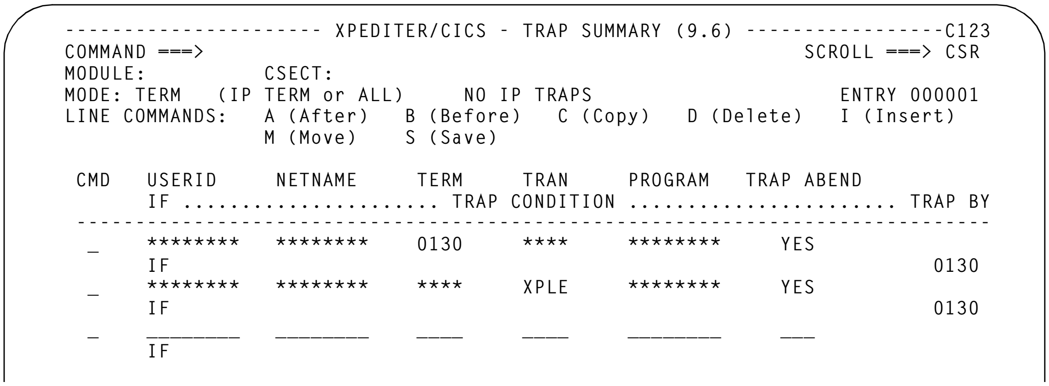
Xpediter/CICS can monitor storage violations in either interactive or unattended mode. In interactive mode, the storage violation is intercepted at either the user's terminal, or a system or help desk terminal. The problem can be reviewed and fixed, or a dump can be requested. In either case, the storage violation is prevented. In unattended mode, the storage violation is intercepted and turned into a harmless abend. Information is passed to the dump dataset with an abend code of ASRA, or to the Abend-AID for CICS Report file with an abend code of STOR.

The following examples show how to set up interactive and unattended monitoring.

Interactive Monitoring

1. Type **XPSP 9.6** and press Enter on a blank CICS screen to display the Trap Summary screen (9.6) shown in Figure 17-7. This screen is used to specify the programs, transactions, userIDs, netnames, and terminals to be monitored for abends. Entries that have your terminal ID in the TRAP BY field will be intercepted by your terminal. You can set traps for Web-based and other transactions using Xpediter's enhanced trap conditions. For more information, see the explanation of the Trap Summary screen (9.6) in the *Xpediter/CICS Reference Manual*.
2. Type **XPLE** in the TRAN field and press Enter to set a trap. The XPLE transaction will be intercepted whenever an abend occurs. See Figure 17-7. If global parameter TRAPTRM is set to YES (the default), your terminal ID will be displayed in the TERM field, and you must overtype it with asterisks (*****).

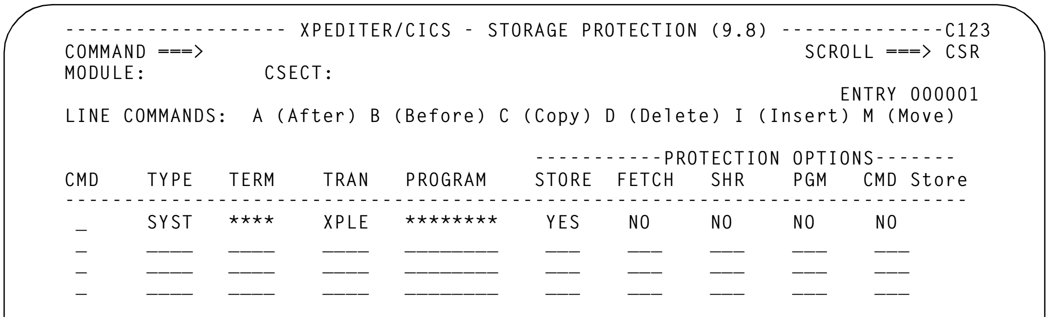
Figure 17-7. Setting an Abend Trap on the Trap Summary Screen (9.6)



```
---------------------- XPEDITER/CICS - TRAP SUMMARY (9.6) -----------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
MODE: TERM   (IP TERM or ALL)     NO IP TRAPS                     ENTRY 000001
LINE COMMANDS:   A (After)   B (Before)   C (Copy)   D (Delete)   I (Insert)
                 M (Move)    S (Save)

 CMD   USERID     NETNAME     TERM     TRAN     PROGRAM   TRAP ABEND
       IF ...................... TRAP CONDITION ....................... TRAP BY
 -------------------------------------------------------------------------------
  _    ********   ********    0130     ****     ********     YES
       IF                                                                0130
  _    ********   ********    ****     XPLE     ********     YES
       IF                                                                0130
  _    ________   ________    ____     ____     ________     ___
       IF
```

3. To set a protection entry, type **=9.8** and press Enter. The Storage Protection screen (9.8) appears as shown in Figure 17-8.

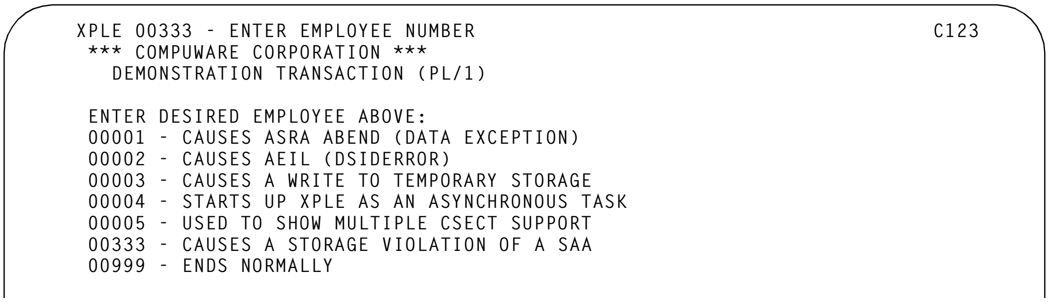
Figure 17-8. Storage Protection Screen (9.8)



```
------------------ XPEDITER/CICS - STORAGE PROTECTION (9.8) -------------C123
COMMAND ===>                                                  SCROLL ===> CSR
MODULE:          CSECT:
                                                                 ENTRY 000001
LINE COMMANDS:  A (After) B (Before) C (Copy) D (Delete) I (Insert) M (Move)

                                        -----------PROTECTION OPTIONS-------
CMD    TYPE   TERM    TRAN   PROGRAM    STORE  FETCH   SHR    PGM   CMD Store
-----------------------------------------------------------------------------
 _     SYST   ****    XPLE   ********    YES    NO     NO     NO     NO
 _     ____   ____    ____   ________    ___    ___    ___    ___    ___
 _     ____   ____    ____   ________    ___    ___    ___    ___    ___
 _     ____   ____    ____   ________    ___    ___    ___    ___    ___
```

4. Type **SYST** in the TYPE field.
5. Type an asterisk (*) in the TERM field.
6. Type **XPLE** in the TRAN field.
7. Type an asterisk (*) in the PROGRAM field.
8. Type **YES** in the PROTECT STORE field and press Enter. This entry means that any storage violation in any program executed by transaction XPLE will be intercepted.
9. Press Clear.
10. Using *another* terminal, sign on to the same CICS region.
11. On a blank CICS screen, type **XPLE** and press Enter.
12. Type **00333** and press Enter. Your terminal will hang as shown in Figure 17-9.

Figure 17-9. Storage Violation on the Demonstration Transaction Screen

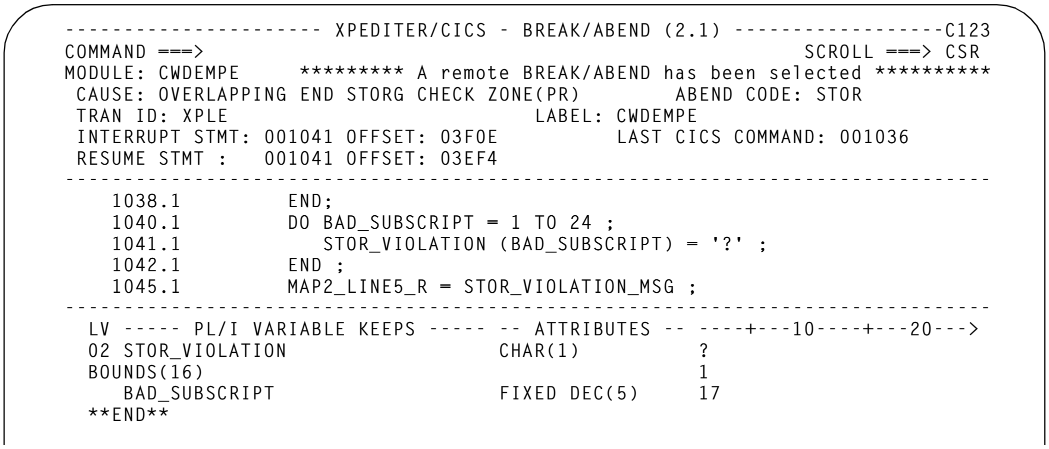


```
XPLE 00333 - ENTER EMPLOYEE NUMBER                                      C123
 *** COMPUWARE CORPORATION ***
   DEMONSTRATION TRANSACTION (PL/1)

 ENTER DESIRED EMPLOYEE ABOVE:
 00001 - CAUSES ASRA ABEND (DATA EXCEPTION)
 00002 - CAUSES AEIL (DSIDERROR)
 00003 - CAUSES A WRITE TO TEMPORARY STORAGE
 00004 - STARTS UP XPLE AS AN ASYNCHRONOUS TASK
 00005 - USED TO SHOW MULTIPLE CSECT SUPPORT
 00333 - CAUSES A STORAGE VIOLATION OF A SAA
 00999 - ENDS NORMALLY
```

Note: The Source Listing screen (2.L) may be displayed in the following step if source support has been turned on for the XPSP transaction.

13. Return to the original terminal. The Break/Abend screen (2.1) will be displayed with a message that a storage violation has occurred, as shown in Figure 17-10.

Figure 17-10. Intercepting a Storage Violation on the Break/Abend Screen (2.1)



```
---------------------- XPEDITER/CICS - BREAK/ABEND (2.1) ------------------C123
COMMAND ===>                                                     SCROLL ===> CSR
MODULE: CWDEMPE     ********* A remote BREAK/ABEND has been selected **********
 CAUSE: OVERLAPPING END STORG CHECK ZONE(PR)        ABEND CODE: STOR
 TRAN ID: XPLE                          LABEL: CWDEMPE
 INTERRUPT STMT: 001041 OFFSET: 03F0E          LAST CICS COMMAND: 001036
 RESUME STMT :   001041 OFFSET: 03EF4
-------------------------------------------------------------------------------
    1038.1         END;
    1040.1         DO BAD_SUBSCRIPT = 1 TO 24 ;
    1041.1            STOR_VIOLATION (BAD_SUBSCRIPT) = '?' ;
    1042.1         END ;
    1045.1         MAP2_LINE5_R = STOR_VIOLATION_MSG ;
-------------------------------------------------------------------------------
  LV ----- PL/I VARIABLE KEEPS ----- -- ATTRIBUTES -- ----+---10----+---20--->
  02 STOR_VIOLATION                  CHAR(1)          ?
  BOUNDS(16)                                          1
     BAD_SUBSCRIPT                   FIXED DEC(5)     17
  **END**
```

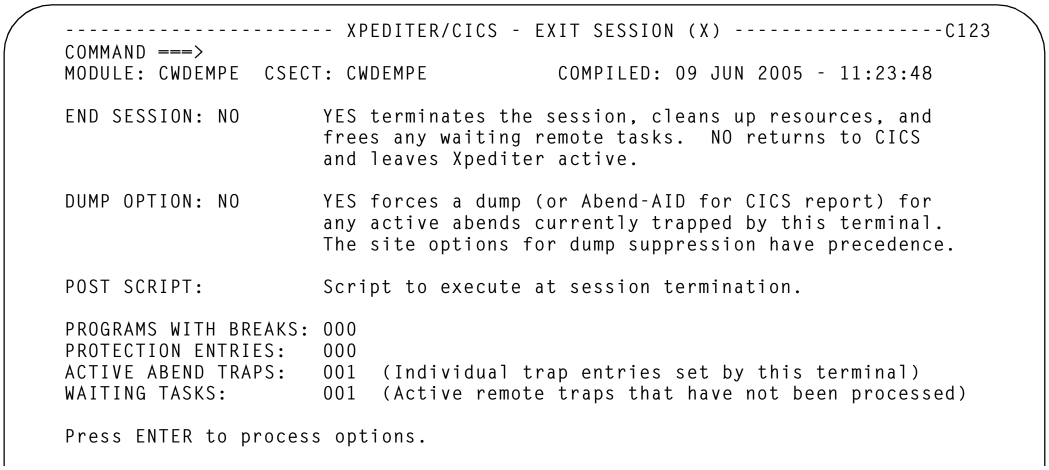
Notice the error message on this screen indicates a STOR violation (OVERLAPPING END STORG CHECK ZONE).

At this point, you have access to all Xpediter/CICS screens and can use them to investigate the problem. In this example, you will end the session.

14. Type **=X** in the COMMAND field and press Enter. The Exit Session screen is displayed as shown in Figure 17-11.

    The ACTIVE ABEND TRAPS and WAITING TASKS fields show the number of remote traps.

Figure 17-11. Removing Abend Trap and the Exit Session Screen (X)



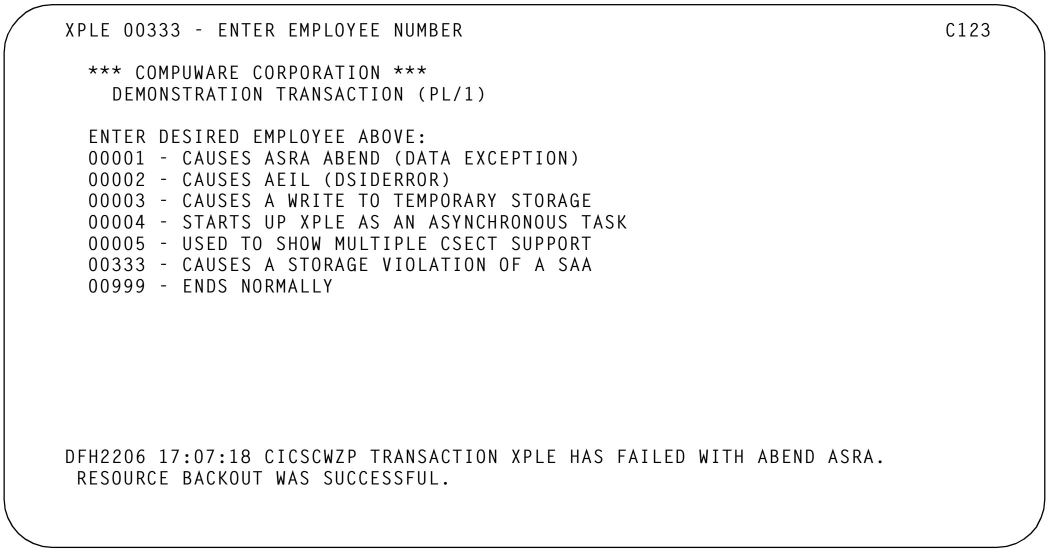
```
----------------------- XPEDITER/CICS - EXIT SESSION (X) ------------------C123
COMMAND ===>
MODULE: CWDEMPE  CSECT: CWDEMPE           COMPILED: 09 JUN 2005 - 11:23:48

END SESSION: NO       YES terminates the session, cleans up resources, and
                      frees any waiting remote tasks.  NO returns to CICS
                      and leaves Xpediter active.

DUMP OPTION: NO       YES forces a dump (or Abend-AID for CICS report) for
                      any active abends currently trapped by this terminal.
                      The site options for dump suppression have precedence.

POST SCRIPT:          Script to execute at session termination.

PROGRAMS WITH BREAKS: 000
PROTECTION ENTRIES:   000
ACTIVE ABEND TRAPS:   001  (Individual trap entries set by this terminal)
WAITING TASKS:        001  (Active remote traps that have not been processed)

Press ENTER to process options.
```

15. Type **Y** in the END SESSION field and press Enter. This frees the remote terminal and removes the trap set on the Trap Summary screen (9.6).

Unattended Monitoring

On the secondary terminal, type **XPLE** and press Enter. The storage violation is intercepted and turned into a harmless ASRA, as shown in Figure 17-12. The dump information is stored on the dump dataset. If Abend-AID for CICS is installed, an abend report is created with an abend code of STOR.

Figure 17-12. Preventing a Storage Violation on the Demonstration Transaction Screen



```
XPLE 00333 - ENTER EMPLOYEE NUMBER                                          C123

  *** COMPUWARE CORPORATION ***
    DEMONSTRATION TRANSACTION (PL/1)

  ENTER DESIRED EMPLOYEE ABOVE:
  00001 - CAUSES ASRA ABEND (DATA EXCEPTION)
  00002 - CAUSES AEIL (DSIDERROR)
  00003 - CAUSES A WRITE TO TEMPORARY STORAGE
  00004 - STARTS UP XPLE AS AN ASYNCHRONOUS TASK
  00005 - USED TO SHOW MULTIPLE CSECT SUPPORT
  00333 - CAUSES A STORAGE VIOLATION OF A SAA
  00999 - ENDS NORMALLY






DFH2206 17:07:18 CICSCWZP TRANSACTION XPLE HAS FAILED WITH ABEND ASRA.
 RESOURCE BACKOUT WAS SUCCESSFUL.
```

Setting Storage Protection during PLT Startup

Storage protection entries can be defined at product initialization. For information on this procedure, refer to the *Xpediter/CICS Installation Guide*.

Chapter 18. Editing CICS Tables and Control Blocks

This chapter discusses how to use the XPSP transaction to perform the following functions:

- Access and modify CICS table entries, storage, and control blocks defined to a CICS region.
- Use Xpediter/CICS to modify CICS tables online without taking the region down.

Note: The demonstrations in this chapter should be performed in Xpediter's standard operating mode — ***not*** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

Editing a CICS Table Entry

The following scenario demonstrates how to edit a CICS table entry on either the Memory Display screen (9.2) or the DSECTs screen (9.D).

Note: This scenario applies to VTAM terminals only.

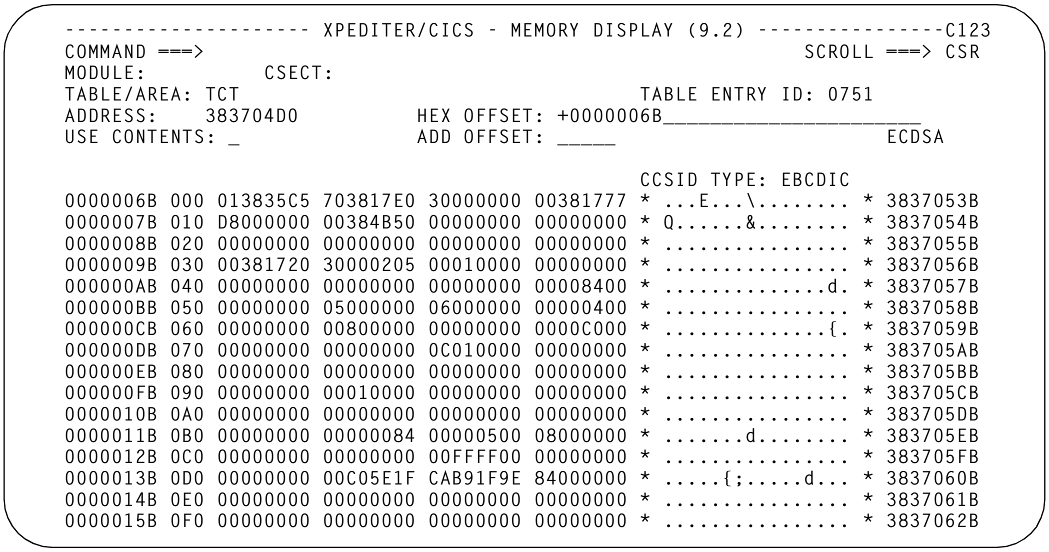
Note: Because the example removes upper case translation by overtyping a field in the TCT, causing CICS not to recognize a lower case transaction name, you may want to just read the example without actually replacing the value.

Editing from the Memory Display Screen

Use the Memory Display screen (9.2) to update tables in hexadecimal dump format:

1. From a blank CICS screen, type **XPSP 9.2** and press Enter. The Memory Display screen (9.2) is displayed (Figure 18-1).

Figure 18-1. Changing the TCT on the Memory Display Screen (9.2)



```
-------------------- XPEDITER/CICS - MEMORY DISPLAY (9.2) ---------------C123
COMMAND ===>                                                 SCROLL ===> CSR
MODULE:          CSECT:
TABLE/AREA: TCT                                  TABLE ENTRY ID: 0751
ADDRESS:    383704D0          HEX OFFSET: +0000006B______________________
USE CONTENTS: _               ADD OFFSET: _____                      ECDSA

                                                 CCSID TYPE: EBCDIC
0000006B 000 013835C5 703817E0 30000000 00381777 * ...E...\........ * 3837053B
0000007B 010 D8000000 00384B50 00000000 00000000 * Q.......&....... * 3837054B
0000008B 020 00000000 00000000 00000000 00000000 * ................ * 3837055B
0000009B 030 00381720 30000205 00010000 00000000 * ................ * 3837056B
000000AB 040 00000000 00000000 00000000 00008400 * ..............d. * 3837057B
000000BB 050 00000000 05000000 06000000 00000400 * ................ * 3837058B
000000CB 060 00000000 00800000 00000000 0000C000 * ..............{. * 3837059B
000000DB 070 00000000 00000000 0C010000 00000000 * ................ * 383705AB
000000EB 080 00000000 00000000 00000000 00000000 * ................ * 383705BB
000000FB 090 00000000 00010000 00000000 00000000 * ................ * 383705CB
0000010B 0A0 00000000 00000000 00000000 00000000 * ................ * 383705DB
0000011B 0B0 00000000 00000084 00000500 08000000 * .......d........ * 383705EB
0000012B 0C0 00000000 00000000 00FFFF00 00000000 * ................ * 383705FB
0000013B 0D0 00000000 00C05E1F CAB91F9E 84000000 * .....{;.....d... * 3837060B
0000014B 0E0 00000000 00000000 00000000 00000000 * ................ * 3837061B
0000015B 0F0 00000000 00000000 00000000 00000000 * ................ * 3837062B
```

2. Type **TCT** in the TABLE/AREA field.
3. Type **TCTEUCTB** in the HEX OFFSET field and press Enter.
4. Look at the data display area. The upper case translation byte value is 01. To turn off upper case translation, position the cursor to the first position displayed, change the 01 to **00**, and press Enter.

   Note: You may want to change the value back to 01 before ending your session.

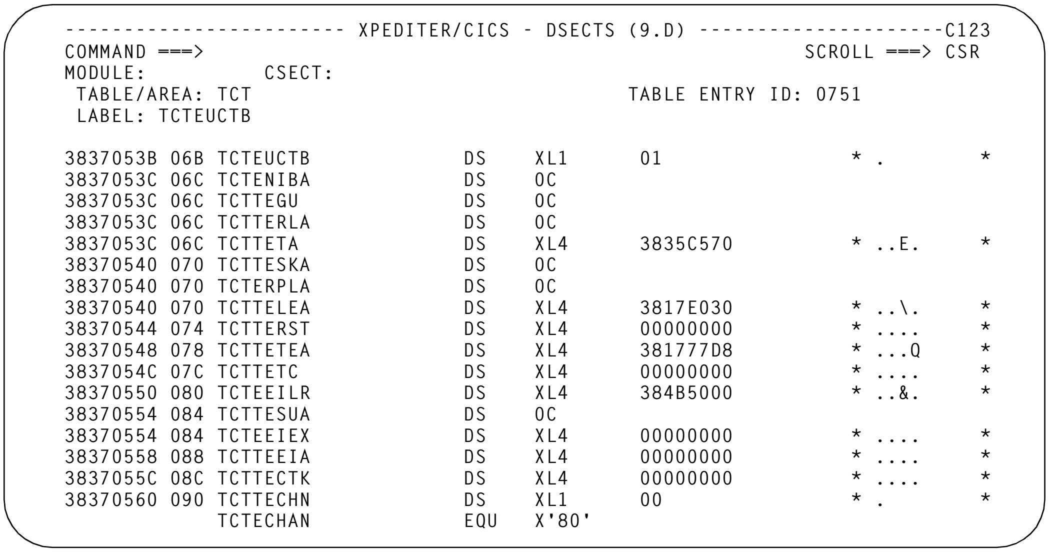
   The table has been updated, and you can now continue your test without recycling the CICS region.

Editing from the DSECTs Screen

You can also edit table entries using the DSECTs screen (9.D). This screen provides a way for you to display and update DSECTs online as shown in the following example:

1. Type **XPSP 9.D** from a blank CICS screen and press Enter.
2. Type **TCT** in the TABLE/AREA field.
3. Type **TCTEUCTB** in the LABEL field and press Enter. The TCTEUCTB field is displayed (Figure 18-2).

Figure 18-2. DSECTs Screen (9.D)



```
------------------------ XPEDITER/CICS - DSECTS (9.D) ----------------------C123
COMMAND ===>                                                   SCROLL ===> CSR
MODULE:          CSECT:
 TABLE/AREA: TCT                               TABLE ENTRY ID: 0751
 LABEL: TCTEUCTB

3837053B 06B TCTEUCTB             DS    XL1      01                * .        *
3837053C 06C TCTENIBA             DS    0C
3837053C 06C TCTTEGU              DS    0C
3837053C 06C TCTTERLA             DS    0C
3837053C 06C TCTTETA              DS    XL4      3835C570          * ..E.     *
38370540 070 TCTTESKA             DS    0C
38370540 070 TCTERPLA             DS    0C
38370540 070 TCTTELEA             DS    XL4      3817E030          * ..\.     *
38370544 074 TCTTERST             DS    XL4      00000000          * ....     *
38370548 078 TCTTETEA             DS    XL4      381777D8          * ...Q     *
3837054C 07C TCTTETC              DS    XL4      00000000          * ....     *
38370550 080 TCTEEILR             DS    XL4      384B5000          * ..&.     *
38370554 084 TCTTESUA             DS    0C
38370554 084 TCTEEIEX             DS    XL4      00000000          * ....     *
38370558 088 TCTTEEIA             DS    XL4      00000000          * ....     *
3837055C 08C TCTTECTK             DS    XL4      00000000          * ....     *
38370560 090 TCTTECHN             DS    XL1      00                * .        *
             TCTECHAN             EQU   X'80'
```

You can change this field (01) by keying over the existing data in either hexadecimal or character format.

This same technique can be used to update any table, control block, or area defined to the CICS region.

Chapter 19.
Using 3270 Web Bridge Support

This chapter demonstrates how to run Xpediter/CICS using the 3270 Web Bridge function. Some typical Xpediter screens are shown, and any minor differences associated with running Xpediter using the 3270 Web Bridge are noted.

You will use the XPED transaction in a web browser window to test the sample application transaction XCB2 and fix an abend. XCB2 is a simple employee payroll transaction that executes the COBOL program CWDEMCB2. The XCB2 transaction is used throughout this chapter to cause several types of abends. (Although COBOL is used in these examples, the same scenarios apply for all languages.)

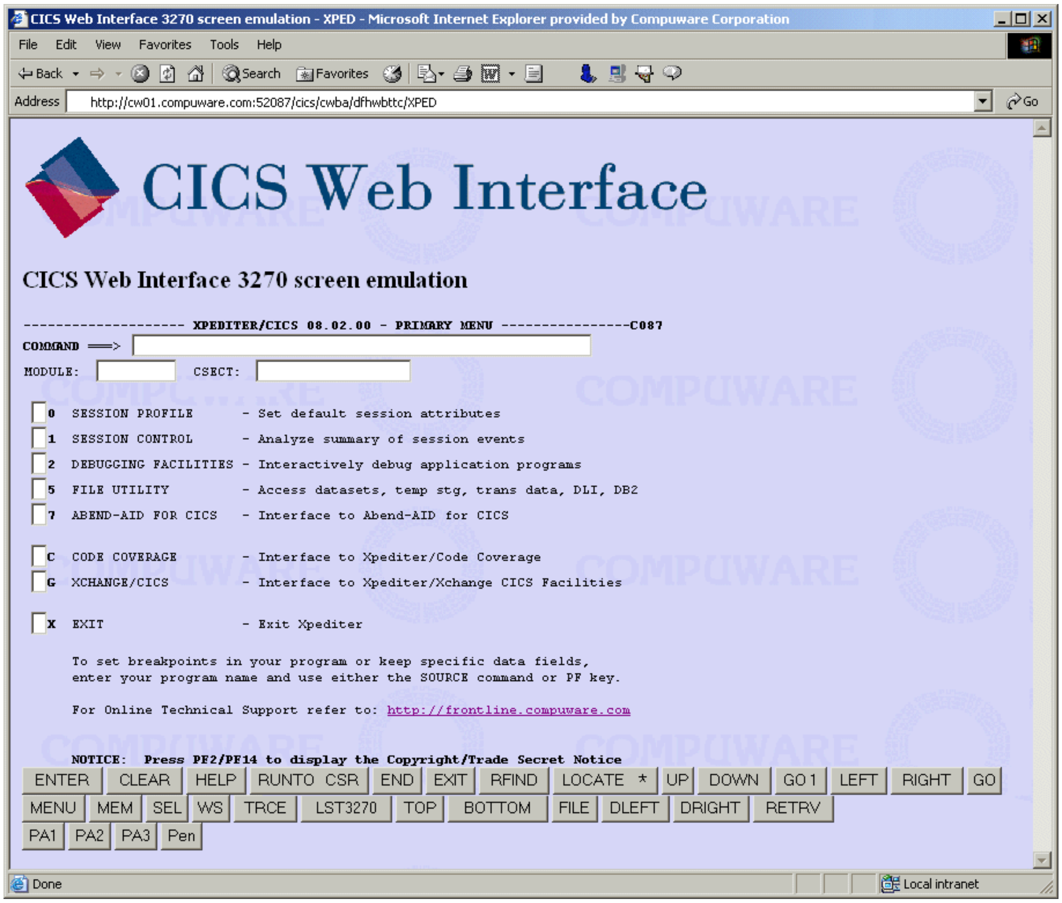
This demonstration assumes your CICS region is configured to allow transactions to be run in a web browser window using the 3270 Web Bridge.

Note: The demonstrations in this chapter should be performed in Xpediter's standard operating mode — ***not*** one of the three restricted modes. For more information, see the *Xpediter/CICS Reference Manual*.

Running Xpediter/CICS Using the 3270 Web Bridge

1. In the Address field of your web browser, type the URL for your CICS region, including the port number, followed by **/CICS/CWBA/DFHWBTTC/XPED** and press Enter. The Xpediter/CICS Primary Menu will appear as shown in Figure 19-1. Input fields appear as text entry boxes, and buttons are provided for standard 3270 terminal keys. Button labels are based on the LABEL column values in your profile.

Figure 19-1. Xpediter/CICS Primary Menu (XPED/XPRT) Using 3270 Web Bridge


CICS Web Interface 3270 screen emulation - XPED - Microsoft Internet Explorer provided by Compuware Corporation
File Edit View Favorites Tools Help
Back Search Favorites
Address http://cw01.compuware.com:52087/cics/cwba/dfhwbttc/XPED
Go
CICS Web Interface
CICS Web Interface 3270 screen emulation
ENTER CLEAR HELP RUNTO CSR END EXIT RFIND LOCATE * UP DOWN GO 1 LEFT RIGHT GO
MENU MEM SEL WS TRCE LST3270 TOP BOTTOM FILE DLEFT DRIGHT RETRV
PA1 PA2 PA3 Pen
Done
Local intranet

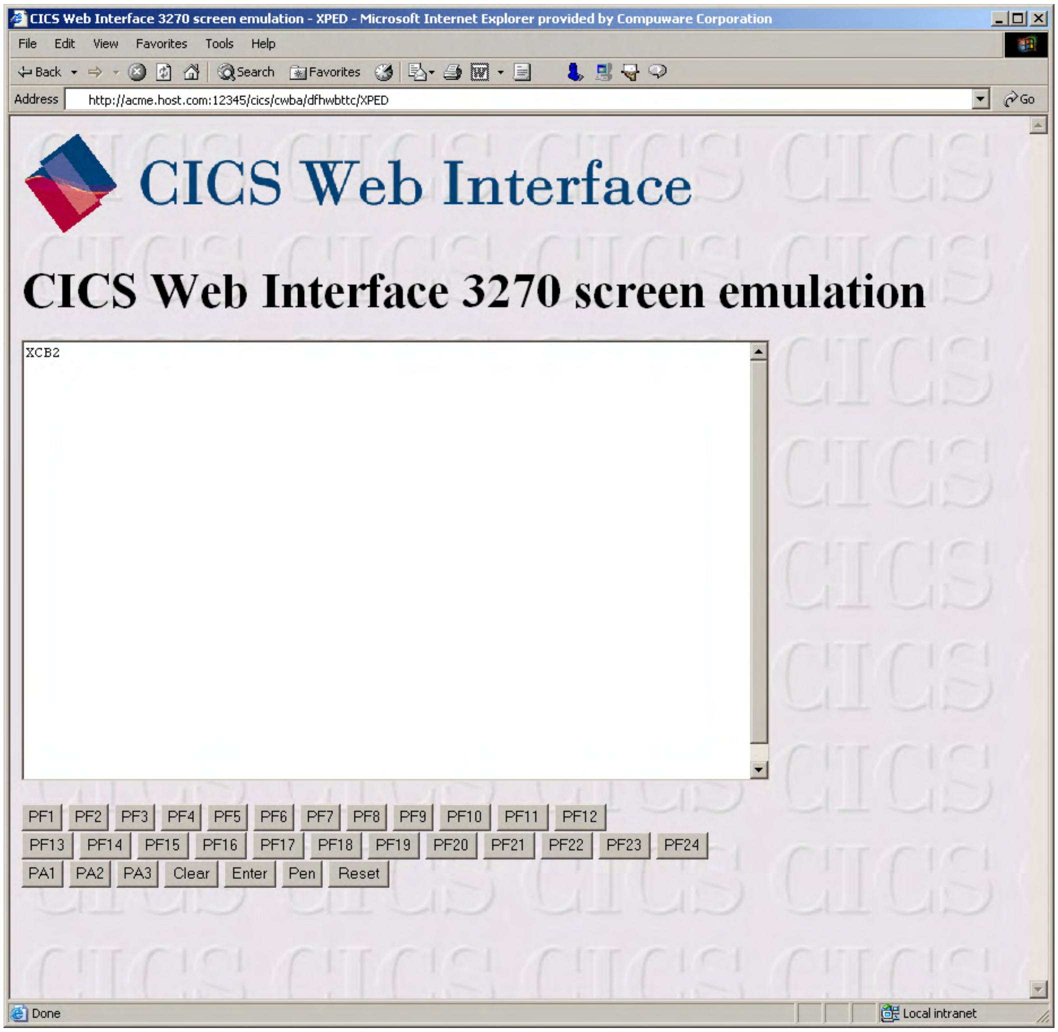
```
------------------- XPEDITER/CICS 08.02.00 - PRIMARY MENU ----------------C087
COMMAND ===>
MODULE:             CSECT:

□ 0  SESSION PROFILE     - Set default session attributes
□ 1  SESSION CONTROL     - Analyze summary of session events
□ 2  DEBUGGING FACILITIES - Interactively debug application programs
□ 5  FILE UTILITY        - Access datasets, temp stg, trans data, DLI, DB2
□ 7  ABEND-AID FOR CICS  - Interface to Abend-AID for CICS

□ C  CODE COVERAGE       - Interface to Xpediter/Code Coverage
□ G  XCHANGE/CICS        - Interface to Xpediter/Xchange CICS Facilities

□ X  EXIT                - Exit Xpediter

     To set breakpoints in your program or keep specific data fields,
     enter your program name and use either the SOURCE command or PF key.

     For Online Technical Support refer to: http://frontline.compuware.com

     NOTICE:  Press PF2/PF14 to display the Copyright/Trade Secret Notice
```

Notes:

- If Xpediter has not yet been initialized in the CICS region, you will be prompted to click the browser Refresh button before the Primary Menu appears.
- The browser progress indicator bar does not complete.
- The browser Back button may yield unpredictable results when using the 3270 Web Bridge.

2. Click the CLEAR button in the browser window. Clicking this button takes the place of pressing the Clear key. A blank CICS screen appears as shown in Figure 19-2.

Figure 19-2. Entering XCB2 on a Blank CICS Screen



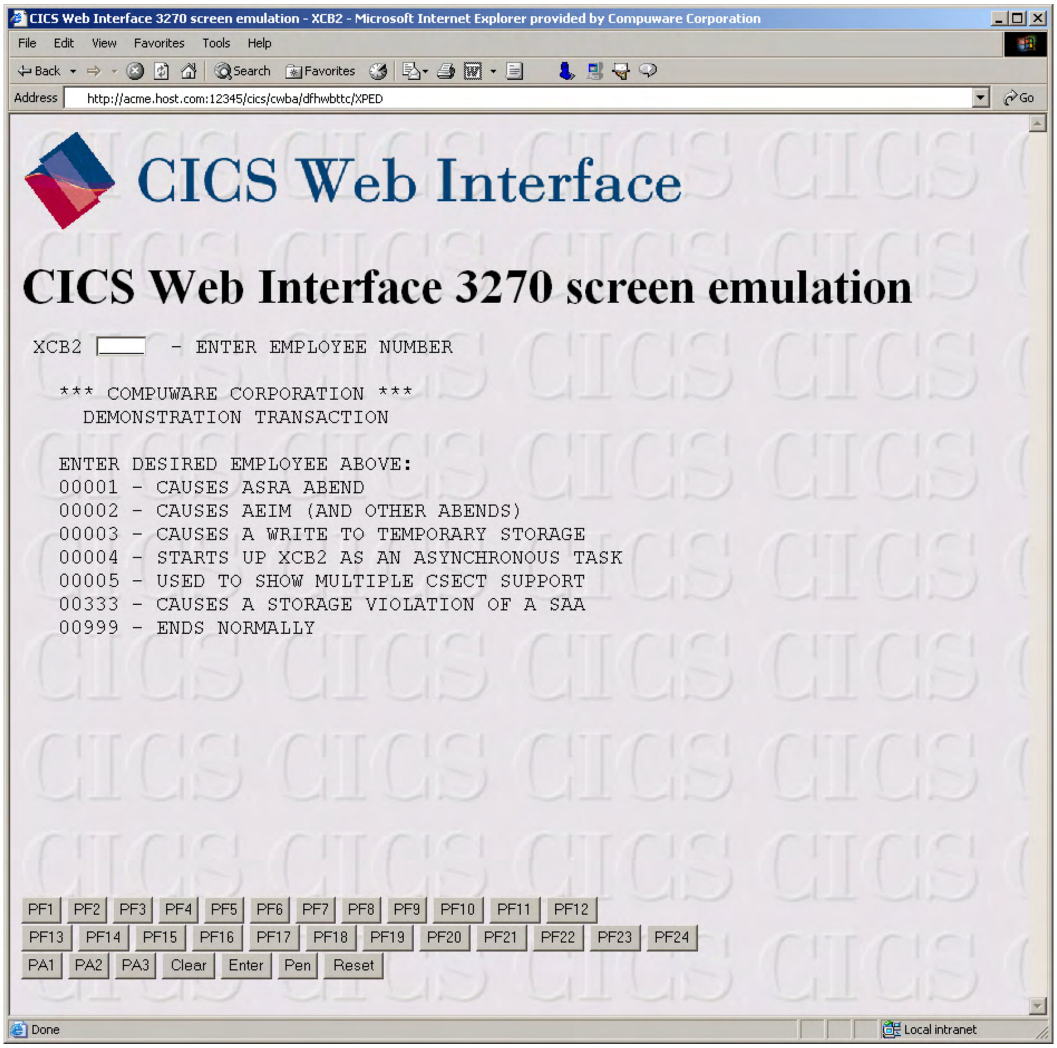
CICS Web Interface 3270 screen emulation - XPED - Microsoft Internet Explorer provided by Compuware Corporation
File Edit View Favorites Tools Help
Back Search Favorites
Address http://acme.host.com:12345/cics/cwba/dfhwbttc/XPED
Go
CICS Web Interface
CICS Web Interface 3270 screen emulation

XCB2

PF1 PF2 PF3 PF4 PF5 PF6 PF7 PF8 PF9 PF10 PF11 PF12
PF13 PF14 PF15 PF16 PF17 PF18 PF19 PF20 PF21 PF22 PF23 PF24
PA1 PA2 PA3 Clear Enter Pen Reset
Done
Local intranet

3. Type **XCB2** in the blank CICS screen and click the Enter button. Clicking this button takes the place of pressing the Enter key. The Demonstration Transaction screen appears (Figure 19-3).

Figure 19-3. Demonstration Transaction Screen



CICS Web Interface 3270 screen emulation - XCB2 - Microsoft Internet Explorer provided by Compuware Corporation
File Edit View Favorites Tools Help
Back Search Favorites
Address http://acme.host.com:12345/cics/cwba/dfhwbttc/XPED
Go
CICS Web Interface
CICS Web Interface 3270 screen emulation

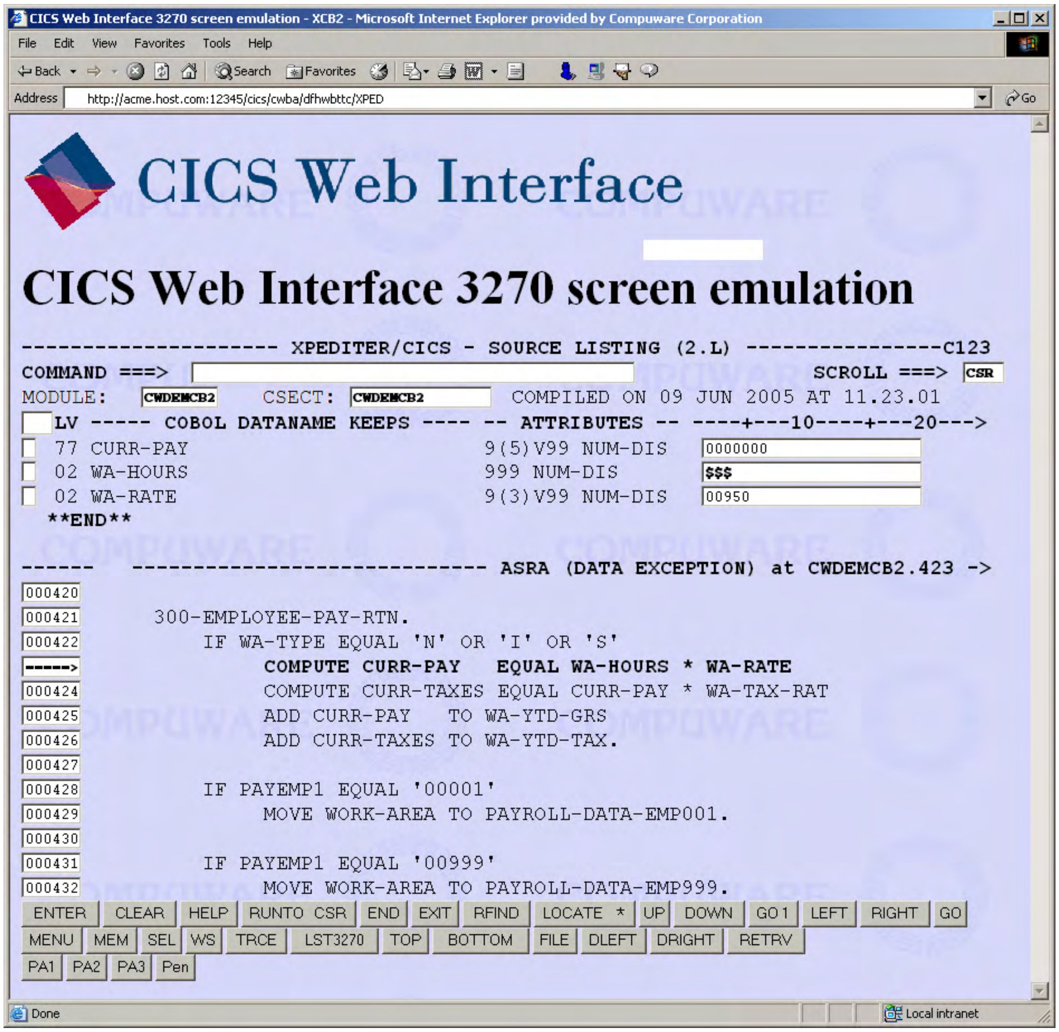
```
XCB2 [     ] - ENTER EMPLOYEE NUMBER

  *** COMPUWARE CORPORATION ***
    DEMONSTRATION TRANSACTION

  ENTER DESIRED EMPLOYEE ABOVE:
  00001 - CAUSES ASRA ABEND
  00002 - CAUSES AEIM (AND OTHER ABENDS)
  00003 - CAUSES A WRITE TO TEMPORARY STORAGE
  00004 - STARTS UP XCB2 AS AN ASYNCHRONOUS TASK
  00005 - USED TO SHOW MULTIPLE CSECT SUPPORT
  00333 - CAUSES A STORAGE VIOLATION OF A SAA
  00999 - ENDS NORMALLY
```

PF1 PF2 PF3 PF4 PF5 PF6 PF7 PF8 PF9 PF10 PF11 PF12
PF13 PF14 PF15 PF16 PF17 PF18 PF19 PF20 PF21 PF22 PF23 PF24
PA1 PA2 PA3 Clear Enter Pen Reset
Done
Local intranet

4. To cause an ASRA abend, type **00001** for the employee number and click the Enter button.

   The CWDEMCB2 demonstration transaction program is intercepted, and the Source Listing screen (2.L) appears as shown in Figure 19-4.

Figure 19-4. Source Listing Screen (2.L) Showing an ASRA


CICS Web Interface 3270 screen emulation - XCB2 - Microsoft Internet Explorer provided by Compuware Corporation
File Edit View Favorites Tools Help
Back Search Favorites
Address http://acme.host.com:12345/cics/cwba/dfhwbttc/XPED
Go
CICS Web Interface
ENTER CLEAR HELP RUNTO CSR END EXIT RFIND LOCATE * UP DOWN GO 1 LEFT RIGHT GO
MENU MEM SEL WS TRCE LST3270 TOP BOTTOM FILE DLEFT DRIGHT RETRV
PA1 PA2 PA3 Pen
Done
Local intranet

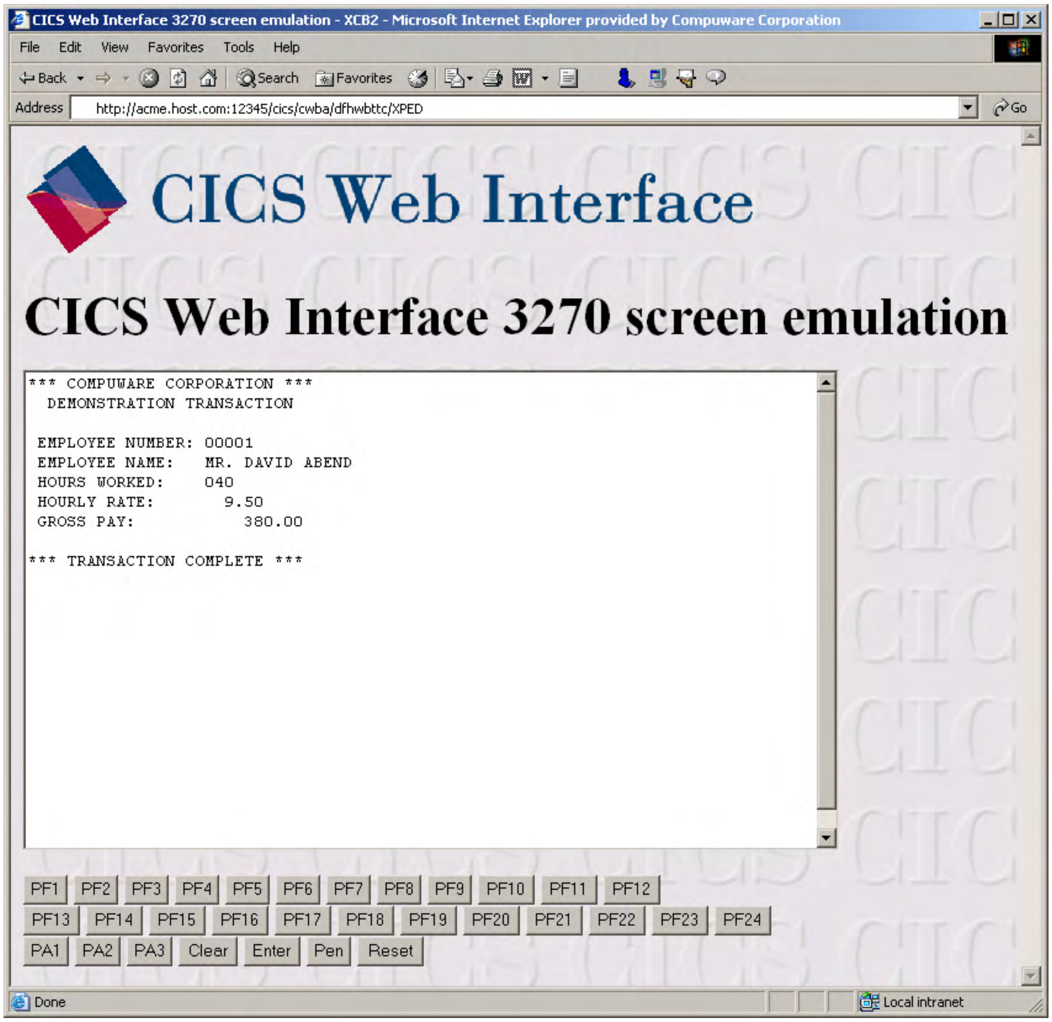
CICS Web Interface 3270 screen emulation

```
--------------------- XPEDITER/CICS - SOURCE LISTING (2.L) ----------------C123
COMMAND ===>                                                 SCROLL ===> CSR
MODULE:  CWDEMCB2     CSECT:  CWDEMCB2       COMPILED ON 09 JUN 2005 AT 11.23.01
□ LV ----- COBOL DATANAME KEEPS ---- -- ATTRIBUTES -- ----+---10----+---20--->
□  77 CURR-PAY                            9(5)V99 NUM-DIS   0000000
□  02 WA-HOURS                            999 NUM-DIS       $$$
□  02 WA-RATE                             9(3)V99 NUM-DIS   00950
   **END**

------  ----------------------------- ASRA (DATA EXCEPTION) at CWDEMCB2.423 ->
000420
000421        300-EMPLOYEE-PAY-RTN.
000422            IF WA-TYPE EQUAL 'N' OR 'I' OR 'S'
----->                COMPUTE CURR-PAY   EQUAL WA-HOURS * WA-RATE
000424                COMPUTE CURR-TAXES EQUAL CURR-PAY * WA-TAX-RAT
000425                ADD CURR-PAY    TO WA-YTD-GRS
000426                ADD CURR-TAXES TO WA-YTD-TAX.
000427
000428            IF PAYEMP1 EQUAL '00001'
000429                MOVE WORK-AREA TO PAYROLL-DATA-EMP001.
000430
000431            IF PAYEMP1 EQUAL '00999'
000432                MOVE WORK-AREA TO PAYROLL-DATA-EMP999.
```

5. Note the value of WA_HOURS. The bad data ($$$) in this field is causing the ASRA. To change it, select the bad data, type **040**, then click the ENTER button.

6. Click the GO button to continue the test. The Demonstration Transaction screen appears as shown in Figure 19-5.

Figure 19-5. Demonstration Transaction Completed



CICS Web Interface 3270 screen emulation - XCB2 - Microsoft Internet Explorer provided by Compuware Corporation
File Edit View Favorites Tools Help
Back Search Favorites
Address http://acme.host.com:12345/cics/cwba/dfhwbttc/XPED
Go
CICS Web Interface
CICS Web Interface 3270 screen emulation

```
*** COMPUWARE CORPORATION ***
  DEMONSTRATION TRANSACTION

 EMPLOYEE NUMBER: 00001
 EMPLOYEE NAME:   MR. DAVID ABEND
 HOURS WORKED:    040
 HOURLY RATE:       9.50
 GROSS PAY:           380.00

*** TRANSACTION COMPLETE ***
```

PF1 PF2 PF3 PF4 PF5 PF6 PF7 PF8 PF9 PF10 PF11 PF12
PF13 PF14 PF15 PF16 PF17 PF18 PF19 PF20 PF21 PF22 PF23 PF24
PA1 PA2 PA3 Clear Enter Pen Reset
Done
Local intranet

7. Remember to end the session as described in "Exiting Xpediter/CICS and Ending a Debugging Session" on page 2-6.

Glossary

AADF. Transaction used to access Abend-AID for CICS to view reports.

abend. Abnormal end of task. The termination of a task, prior to normal completion, due to an unresolved error condition.

Abend-AID for CICS. System software product designed to analyze the cause of CICS transaction abends. It provides online diagnostics that explain the cause of the abend, the location within the program where the error occurred, and how to correct the problem.

abend code. Identifier for the current abend when an abend occurs.

Abend trap. Function that allows Xpediter/CICS to notify the user when a program breakpoint, abend, or storage violation has been encountered. See "trap."

ADD OFFSET. Indicates a specific screen location. The data at this location is scrolled to the top of the data portion of the screen. If the USE ADDRESS field contains an "X", Xpediter/CICS uses the data at this location as an address and displays the location indicated. This feature is found on screens 2.2 and 9.2.

ADDRESS. Address that is displayed on hexadecimal dump format screens and indicates the beginning address of the area being accessed. The address does not change unless:

- A new area is accessed
- The programmer types over the address with a location to be accessed
- The USE CONTENTS field is used to retrieve data from a stored location.

Assembler language processor. One of several language processors provided by Compuware, this language processor accepts Assembler output, builds sort work records, sorts and merges the records, and merges the records with the listing to produce processor control blocks that can then be used as input to Xpediter/CICS and other Compuware products.

Basic Direct Access Method (BDAM). File access method that directly retrieves or updates specified blocks of data on a direct access storage device.

Basic Partitioned Access Method (BPAM). File access method that can be applied to create program libraries, in direct access storage, for convenient storage and retrieval of programs.

batch. Processing in which jobs are grouped (batched). The jobs are executed sequentially, and each job must be processed to completion before the following job can begin execution.

BDAM. Basic Direct Access Method.

BK-CHAIN. Back chain pointer.

blank. Part of a data medium in which no characters are recorded. Its hexadecimal character representation is X'40'.

BL cell. Base Locator cell. These cells are used by COBOL programs to provide addressability to data within the working storage section of a program.

BLL cell. Base Linkage Locator cell. These cells are used by COBOL programs to provide addressability to data within the linkage section of a program.

BLW cell. Base Locator Working Storage cell. These cells are used by COBOL II programs to provide addressability to data within the working storage section of a program.

BPAM. Basic Partitioned Access Method.

breakpoint. Xpediter/CICS method of pausing program execution during debugging. Setting a breakpoint in a program halts execution of the program when a statement is about to be executed. Breakpoints may be set as conditional or unconditional and to occur either before or after the event.

At a breakpoint, the user may view all the tables and areas necessary to solve a problem. After interrogating the problem, the user may continue normal execution of the system.

channel. Analogous to a parameter list. Containers are grouped together in sets called channels.

CICS. Customer Information Control System.

C language processor. One of several language processors provided by Compuware, this language processor accepts C output, builds sort work records, sorts and merges the records, and merges the records with the listing to produce processor control blocks that can then be used as input to Xpediter/CICS and other Compuware products.

CLC. Compare Logical Character.

CLOT. (DB2 only) CICS Life-of-Task block.

COBOL language processor. One of several language processors provided by Compuware, this language processor accepts COBOL compiler output, builds sort work records, sorts and merges the records, and merges the records with the listing to produce processor control blocks that can then be used as input to Xpediter/CICS and other Compuware products.

command. Request from a terminal to perform an operation or to execute a program.

COMMAND field. Field that appears in the upper left corner of most Xpediter/CICS screens. All Xpediter/CICS functions and screens are selected by entering the desired function or screen ID in the COMMAND field.

Compare Logical Character (CLC) instruction.

An IBM machine compare instruction.

Compuware Shared Services (CSS). A set of components used by several Compuware products to provide storage, retrieval, and maintenance for source listings and abend reports.

Conditional breakpoint. For a statement set with a conditional breakpoint, Xpediter/CICS interrogates the condition entered and if true, halts execution of the program and receives control.

Containers. The “big commarea” containers and channels that IBM introduced in CICS Transaction Server 3.1 to replace commareas. For additional information, refer to IBM’s “CICS Information Center”.

CSECT field. Field on most Xpediter/CICS screens that is used to change the CSECT to be accessed during the session.

dataname. The name of the data item in the working storage section of a COBOL program.

dataset. Collection of data treated as a unit that is the primary unit of access and storage. It can be organized in various ways.

DA-KEY. BDAM blocked DAtaset, deblocked by KEY.

DA-REL. BDAM blocked DAtaset, deblocked by a RELative record.

Database Descriptor (DBD). Defines the database associated with a PCB. Associated with every PSB is a list of PCBs that define databases that can be accessed via a PSB. Each PCB-accessed database can be identified by its DBD name, which is used during the definition of the database to DL/I.

DA-UNB. Direct Access UNBlocked dataset.

DB2. An IBM relational database management system.

DBD. DataBase Descriptor.

DBPA. Transaction used to initiate certain Xpediter/CICS functions at system start-up.

DCT. Destination Control Table.

DDIO. A Compuware file access method.

DDIO file. A generic name for an Abend-AID report file, or an Xpediter source listing file.

DDIO file member. A generic name for an abend report in an Abend-AID report file, or an Xpediter source listing in a source listing file.

default value. Choice among exclusive alternatives made by the system when no explicit choice is made by the user.

destination. Location at which a block of (lines) records or a single record is being copied or moved. The destination can be specified with an A (After) or B (Before) line command.

Destination Control Table (DCT). A CICS table describing transient data files.

DFHRPL. Library of CICS and user-written programs that are loaded and executed as part of a CICS system. Programs from this library are loaded into CICS storage for online execution. DFHRPL may be supplemented by LIBRARY resource definitions in CICS TS 3.2 and above.

DL/I. Data Language 1.

DMAP. Data division map of a COBOL program. The DMAP COBOL compile option produces a report of all datanames and their associated BL or BLL cell, displacement within the cell, and field description.

DOS. Disk Operating System.

DSECT. Dummy control section. A control section that an Assembler program can use to format an area of storage without producing any object code.

EDT. MVS Eligible Device Table.

EIB. Execute Interface Block.

EIS. Execute Interface Storage.

entry-sequenced dataset (ESDS). VSAM dataset whose records are loaded in sequence. Unlike a normal sequential dataset, ESDS records can be accessed randomly by their addresses.

ESDS. Entry-Sequenced Dataset.

external security manager (ESM). A security product that provides an interface to the z/OS System Authorization Facility (SAF). CA-ACF2, RACF, and CA-Top Secret are examples of ESMs.

Execute Interface Block (EIB). Block that contains information pertinent to a command-level transaction such as the current time and date, transaction ID, task number, terminal ID, COMMAREA length, attention identifier, function code, and response code.

FCT. File Control Table.

file. Complete organized collection of information.

File Control Table (FCT). A CICS table defining files that can be accessed by CICS programs. Also used to generically apply to files defined in the CSD.

HELP. Primary command that requests Xpediter's interactive Help facility.

HEX. Primary command that alternates between symbolic or dump format display.

HEX OFFSET. Hexadecimal value of the location of the retrieved area relative to the beginning of the address. The sum of the ADDRESS field and HEX OFFSET field indicates the actual address of the data displayed.

HEX ON and HEX OFF. Displays data in character and zoned decimal format on the VARIABLE STORAGE screen (2.3). The user may view this same data in a standard hexadecimal dump format by entering HEX ON in the COMMAND field. HEX OFF redisplays the data in character and zoned decimal format.

hung task. A task suspended by the system.

hung task analysis. A Xpediter/CICS facility for the XPSP user that allows interrogation of a suspended task to determine the reason for its suspension. This function is available by using the VIEW SINGLE TASK screen (9.1) and the LIST ALL TASKS (9.3) screen.

IMS. Information Management System.

INRWORK. INput Register Work area.

interactive. Pertaining to an application in which each entry calls forth a response from a system or program.

IP address. A numeric address given to servers and users' computers connected to the Internet.

ISPF. Interactive System Productivity Facility.

JCL. Job Control Language.

key. Code used to locate a record and establish its position in an index. The key can be part of a field, a full field, or multiple fields duplicated from the record.

Key-Sequenced Dataset (KSDS). VSAM file type whose records are loaded in key sequence. Records are retrieved by key or address using an index. New records are inserted in key sequence by means of distributed free space.

keyword. Reserved word that has special significance.

KSDS. Key-Sequenced DataSet.

line command. Edit command that is entered directly on the line to be processed by overtyping the sequence number at the beginning of the line. Also known as a prefix command.

linkage section. A section of a COBOL program used to describe data that is passed to it from CICS or another program.

MENU. Primary command that ends the current function and returns the user to the SYSTEM MENU.

MODULE field. Field on most Xpediter/CICS screens that is used to change the program to be accessed during the session.

MORE. Indicates there is more information to display. This indicator appears only when screen overflow data is not generally assumed. Use PF7 and PF8 to view the additional data.

MQ. IBM licensed programs that provide message queuing services.

offset. A relative location or position within a data area.

OFL. Optional Features List. Also known as OPFL.

operating system. Software that controls the execution of jobs. It may provide resource allocation and scheduling.

OS. Operating System.

paragraph. Set of one or more COBOL sentences, making a logical processing entity, and preceded by a paragraph name or a paragraph header.

PCB. Program Communication Block.

PLIST. (DB2 only) Parameter List.

PF key. Program Function Key.

PL/I language processor. One of several language processors provided by Compuware, this language processor accepts PL/I compiler output, builds sort work records and an incore symbol table of all the identifiers, and produces processor control blocks that can then be used as input to Xpediter/CICS and other Compuware products.

PLT. Program List Table.

prefix commands. Another name for line commands.

Program Communication Block (PCB). One of a list of control blocks used by DL/I that define the databases that can be accessed via a particular PSB. Each PCB-accessed database DBD name. For each PCB with a PSB, Xpediter/CICS displays the PCB number and DBD name. Xpediter/CICS uses the PCB number to identify the PCB to access a DL/I database from the selected PSB.

Program List Table (PLT). CICS table describing a list of programs to be executed when CICS is in initialization or termination processing.

Program Specification Block (PSB). DL/I control block that defines a set of DL/I databases that can be accessed from a program. The databases and segments in the databases that can be accessed are defined via a list of PCBs defined in the PSB. To access any DL/I database, always select a PSB.

program storage. Class of CICS storage used for application programs.

primary command. Command that provides a general function. Primary commands are entered in the COMMAND field.

procedure division. Section of a COBOL program that contains executable instructions.

profile. Control block that defines session characteristics applicable to one or more users of Xpediter/CICS. Pertinent information in a profile may include PF key settings, default abend trap, trace, storage protection, and footing settings.

profile dataset. VSAM KSDS dataset containing all profiles available for use during a Xpediter/CICS session.

program function (PF) key. Keyboard keys that are numbered from PF1 to PF24 and are programmed to perform functions such as scrolling.

Program Status Word (PSW). A special control register, in the hardware, defining the current status and location of a program that is executing.

PSB. Program Specification Block.

PSW. Program Status Word.

quick table disable. Method of turning off storage protection monitoring without destroying the current table entries.

RCT. (DB2 only) Resource Control Table.

record. Collection of related data or words treated as a unit.

register. Storage device, having specified storage capacity such as a bit, byte, or a computer word, and usually intended for a special purpose.

register save area. Group of 72 contiguous bytes used for saving registers when one program calls another.

Relative Record DataSet (RRDS). VSAM dataset whose record locations are specified by a number that represents a record's location in the dataset relative to the beginning of the dataset.

remote task trapping. Xpediter/CICS function that allows one terminal to trap abends, breakpoints, and storage violations that occur on another terminal or in a non-terminal task.

RRDS. Relative Record Dataset.

screen ID. Code entered in the COMMAND field to retrieve information and function screens. The SCREEN ID for each screen appears on every screen immediately preceding the screen title.

SCROLL field. A field on most Xpediter/CICS screens that is used to set the default value to be used for those screens that allow scrolling.

Segment Search Argument (SSA). A control block used by DL/I to access a segment within the hierarchy of a database.

Shared Directory. A variable-length record VSAM RRDS that maintains information about abends and language processing along with the attached database activity. A shared directory can contain Abend-AID for CICS directory records for each region and transaction dump known to a server, Abend-AID directory records for abend report processing, or source listing shared directory records necessary to process source listing database members.

shared storage. Class of CICS storage that can be shared between tasks.

SIT. System Initialization Table.

snap dump. Dump that is taken at a specific point during execution of a program. Processing is generally continued after the dump has been taken.

SQLCA. (DB2 only) SQL Communication Area.

SQLDA. (DB2 only) SQL Descriptor Area.

SSA. Segment Search Argument.

statement number. Sequence numbers provided by compilers and assemblers to provide the programmer with an easy means of identifying a statement within a program.

statement number column. Column on the display where the statement numbers are located. In some cases, commands may be entered into this column.

STCA. System portion of the Task Control Area.

stop. See breakpoint

storage protection. Method of preventing programs from violating storage within the CICS region, thereby increasing the time that the region remains running.

storage protection exceptions. Xpediter/CICS facility for allowing specific actions that are considered to be storage violations to take place.

system labels. Labels that equate to storage locations that may be used by any Xpediter/CICS user. Some system labels are predefined by Compuware.

table entry ID. Identification command used to directly retrieve the desired CICS table entry for display.

task. Execution of a program or multiple programs within CICS to perform a specific function. Each task is assigned a unique number (task number) by CICS.

TCA. Task Control Area.

TCP/IP. Transmission Control Protocol/Internet Protocol. Set of communication protocols enabling Telnet, FTP, e-mail, and other services.

TCT. Terminal Control Table.

Terminal Control Table (TCT). CICS table defining terminals used by CICS.

TOGGLE. Primary command that transfers you to the Source Listing (2.L) screen.

trace. Record of the execution of a computer program; it exhibits the sequences in which the instructions were executed.

trace table. CICS storage area into which trace information is placed. This table contains the chronological occurrences of events that take place in CICS, recorded in wraparound fashion within the trace table.

trap. Xpediter/CICS feature. The user may set traps to intercept CICS transaction abends and to view all areas at abend time. Traps may be set to only intercept transaction abends at terminals running Xpediter/CICS or designated as remote. Traps may be set to intercept specific transaction's abends, regardless of the terminal from where they are executed. Traps may also be used to intercept abends occurring in non-terminal tasks.

TSA. Temporary Save Area.

unconditional breakpoint. For a statement set with an unconditional breakpoint, Xpediter/CICS receives control and temporarily halts execution of the program before or after this statement.

use address. A feature that is available on the Memory Display screens (2.2 and 9.2). The USE ADDRESS function automatically retrieves an address reference from memory and displays the location.

user labels. Labels generated by a user that equate to storage locations.

variable. Name of a data item in a PL/I program.

view. Xpediter/CICS method of selecting data items for display at a breakpoint or abend.

virtual storage. Storage space that may be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses.

Virtual Storage Access Method (VSAM). File access method whereby the records in a file on a direct access storage device can be accessed in key-sequence (KSDS), entry-sequence (ESDS), or relative record sequence (RRDS).

VSAM. Virtual Storage Access Method.

VTAM. Virtual Telecommunications Access Method.

working storage. A section of a COBOL program used to define the data items that are used in a program.

XPED. Transaction code entered to invoke Xpediter/CICS. It provides a source-based focus for debugging CICS application programs.

XPND. Transaction code entered to end an Xpediter/CICS debugging session.

XPRT. Transaction code entered to invoke Xpediter/CICS. XPRT provides a break/abend focus for debugging CICS application programs.

XPSP. Transaction code entered to invoke the XPSP level of Xpediter/CICS. This level is to be used only by experienced system programmers authorized to update CICS tables and control areas. System-wide storage protection is set by the XPSP user. This is the only Xpediter/CICS level that does not restrict updating.

Index