



Track down  
your CICS  
program bugs with  
**TRACK**



**TRACK** provides on-line testing and debugging capabilities in CICS environments. It increases CICS reliability and improves productivity by enabling application programmers to detect and correct multiple errors in a single debugging session.

## ***TRACK provides***

### Year 2000 Testing

**TRACK** accepts a future run date (i.e. 2000/01/01) for an individual debug session. This allows testing selected programs **WITHOUT** affecting other transactions and programs, **WITHOUT** requiring an LPAR or an IPL, and **WITHOUT** affecting the rest of the CICS region.

### Faster Debugging

**TRACK** provides faster debugging of program logic, coding, and data errors. It provides for user halts at the transaction level or in any subroutine and halt of program execution at programmer defined halt points which are activated only when certain conditions are encountered. Data can be displayed by COBOL dataname. Debugging of tasks executing at another terminal or printer or unattached to any terminal can be done. Loops can be trapped by setting instruction and CICS call limits.

### Faster Corrections

**TRACK** allows single stepping through a program to follow the program logic. This can be done by either single machine instruction or by program statement. It also pinpoints errors for you on the screen. COBOL, PL/1 and Assembler source can then be displayed as well as data files. Corrections are made interactively and execution of the program then continues. Program flow after a halt can be redirected to test infrequently used logic paths.

### Faster Turn Around

Multiple errors can be examined and resolved in one execution of the program. No more waiting to resubmit the program for additional compiles, doing more testing, and looking at more dumps.

### Reduced Dump Analysis

It is no longer necessary to plow through core dumps. **TRACK** points out exactly what you need to know on the screen. It takes the drudgery out of debugging and testing.

## Stable CICS Environment

**TRACK** can also be used to monitor any specific program running under CICS. It detects program abends or illegal CICS operations. **TRACK** protects CICS from transactions causing table storage violations, thus preventing CICS crashes. **TRACK** uncovers those intermittent, hard to find bugs, in both a testing and production environment.

## Security

A powerful security feature enables the system administrator to control both who may use the **TRACK** system and which facilities are available to each individual. An audit trail of all alterations made using **TRACK** can be obtained.

## Benefits you will see

- Improved programmer productivity from CICS test sessions

- Reduced development times

- Reduced requirement for tedious dump analysis

- Ability to test infrequently used logic paths to identify and correct obscure program errors

- Better tested and more resilient programs

- Improved CICS system stability

## Easy to use

**TRACK** is extremely easy to use. It is a menu and PF key driven system with on-line help screens. All facilities are accessible from clear informative menus, by use of commands or via 'fastpath' identifiers. The interactive screens that highlight error conditions are easy to understand.

**TRACK** is used by application programmers to locate logic errors, coding errors, and data errors in programs.

**TRACK** is used by the system programmer to locate conditions that cause CICS crashes, including invalid modification of CICS tables.

**TRACK** is an outstanding learning aid for junior programmers. They can easily follow program logic, learn CICS conventions and standards, and get around the problems always encountered by new programmers.

## Menus and Fastpaths

Menu screens giving access to all the main facilities of **TRACK** are available throughout the system. They provide an indication of the functions available and allow easy use of them. The following example of the highest level menu shows how they work.

At the top of the screen are three fields seen throughout the system, **Command**, **Offset**, and **Password**.

You may obtain help information relating to a topic by position-ing the cursor at the required option and pressing the **HELP** key.

```
=0 TRACK V62020 - Primary Menu CICSUSER T08A 01/14/98 08.49.03
Command-----> -
Offset-----> Password----->

1 - Display Program Source, set Halt Points (=1)
2 - Display Program Structure (=2)
3 - Display/Alter Storage Areas (=3)
4 - Display/Alter CICS Areas (=4)
5 - Current Halt Status (=5)
6 - Continuation Options (=6)
7 - Halt Point Management (current task) (=7)
8 - Monitor Controls (=8)
F - File Facility (=F)
Q - Temporary Storage Facility (=Q)
R - Program Residency (=R)
S - Session Controls (=S)

Type selected option code in the command field and press the ENTER key
or press one of the listed function keys

1=HELP 3=EXIT 6=? 9=ZOOM 12=END
```

**Command** is the command input field into which may be typed any valid **TRACK** command. When used at a menu, the option selected may be entered here. You may also enter a 'fastpath' id here at any time to take you directly to a particular menu or function screen.

On menus you may select an option by typing its identifier in the command field. Alternatively, you may position the cursor at the required line and press the **ZOOM** PF key. The values shown in brackets to the right of each option are the fastpath identifiers associated with each activity. They may be used at this or any other screen for direct access to a facility.

For example, to select the Monitor Controls function you could supply either the option value '=8' or if you know the screen you want, you could be more specific (=8.2, for instance). Alternatively, you could position the cursor anywhere on the line containing option 8 and press the **ZOOM** PF key.

**Offset** allows you to input a value to modify the effect of certain commands. For example, if an area of storage is to be displayed, the entered value determines the offset within the area at which the display will begin.

**Password** is the field where any password required by the system for the function being performed must be supplied.

## Standard Halt Points

**TRACK** will halt a monitored program's execution at user-specified locations. When the intercept occurs, a halt screen is displayed, including the relevant source code if available. All **TRACK** debugging facilities are then available for the examination and alteration of data (in storage, temporary storage, or on file) and of program machine code. In this way errors may be found and corrected or changes made to ensure execution of particular sections of code. After each halt the programmer may terminate the transaction, produce a dump, continue execution normally, **STEP** through the program by statement or instruction, or continue from a different point in the program.

Standard halt points can be set by specifying one of the following:

- 1) **START** which indicates to halt at program entry point,
- 2) a particular program statement which causes a halt immediately before the statement is executed, and
- 3) a 1 to 6 character even hexadecimal offset (whose value must not exceed the size of the program or module specified).

## Display/Debug Facility

Halt points may be specified for a program to enable controlled interruption of program execution so the contents of program variables, etc. may be examined. The reason for the interruption is given including the name of the program interrupted, the reason for the interruption, the offset in the program at which the interruption occurred, the machine instruction at the interrupt point, and the values of the program registers. Whenever a halt point is reached the resulting screen highlights the statement for which the halt occurred.

The Monitor Control screen is completed by entering NWSP07 in Program Name and START in Halt Offset. This directs TRACK to monitor program NWSP07 and halt at program start.

To simulate execution on January 1st, 2000, the run date is set to 2000/01/01. Program requests for CURRENT-DATE or EIBDATE will return this date. Date Access Action of HALT causes Track to automatically halt each time a system date is accessed.

```
=8.2          TRACK - Monitor Controls      CICSUSER T08A 01/14/98 10.31.13
Command----->
Offset----->                                Password----->

                                Add Monitor Controls
Terminal id--->                                Task id----->
Program Name--> nwsp07                          Module Name--->

Halt Offset---> start
Times to Halt->
1st Halt Pass->
Pass Increment>
Halt Condition>

Storage Protection->

Run Date-----> 2000/01/01
Date Access Action-> halt_

1=HELP 3=EXIT 6=? 12=END
```

To also set a halt each time statement 921 is reached, enter S921 in Halt Offset.

```
=8.2          TRACK - Monitor Controls      CICSUSER T08A 01/14/98 11.44.32
Command----->
Offset----->                                Password----->

                                Add Monitor Controls
Terminal id--->                                Task id----->
Program Name--> nwsp07                          Module Name--->

Halt Offset---> s921
Times to Halt->
1st Halt Pass->
Pass Increment>
Halt Condition> _

Storage Protection->

Run Date----->
Date Access Action->

1=HELP 2=STOP 3=EXIT 4=STEP 6=? 11=FLOW 12=CONT
```

```

=5.1          TRACK - Stop Display          CICSUSER T08A 01/14/98 11.30.27
Command-----> _
Offset----->
Password---->

Stop Program--> NWSP07   Current Module-> NWSP07   At Terminal---> T08A
Phase Offset--> 0009DC   Module Offset--> 0009BC   Statement No.-> 000710
Stop Reason---> HALT REQUEST - HALT ID = 1
  Stmt  ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
000707 029200
000708 029300 PROCEDURE DIVISION USING DFHEIBLK DFHCOMMAREA.
000709 029400 0000-INITIALIZATION-AND-CONTROL.
000710 029500 IF EIBCALEN GREATER ZERO
-----
OP1--> EIBCALEN
00000000 01F4                                500          Type-> Binary
                                           001840E8
-----
000711 029600 MOVE DFHCOMMAREA TO CA-AREA
000712 029700 ELSE
000713 029800 GO TO 9999-END.
000714 029900
000715 030000 IF CA-PROGRAM-ID EQUAL 'P07'
000716 030100 GO TO 1000-MAIN-MENU.

1=HELP 2=PCK 3=EXIT 4=STEP 6=? 7=BWD 8=FWD 9=ZOOM 10=WINDOW 11=FLOW 12=CONT

```

When entering program NWSP07 this screen is displayed with the first executable statement highlighted since a halt was set at program START.

If you press PF12, program execution of NWSP07 continues until it encounters the halt set for statement 921. Statement 921 is then displayed in context with statement 921 highlighted.

```

=5.1          TRACK - Stop Display          CICSUSER T08A 01/14/98 11.31.49
Command-----> _
Offset----->
Password---->

Stop Program--> NWSP07   Current Module-> NWSP07   At Terminal---> T08A
Phase Offset--> 0009EC   Module Offset--> 0009CC   Statement No.-> 000711
Stop Reason---> STEP REQUEST
  Stmt  ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
000708 029300 PROCEDURE DIVISION USING DFHEIBLK DFHCOMMAREA.
000709 029400 0000-INITIALIZATION-AND-CONTROL.
000710 029500 IF EIBCALEN GREATER ZERO
000711 029600 MOVE DFHCOMMAREA TO CA-AREA
-----
OP1--> DFHCOMMAREA
00000000 F040D5E6 E2D7F0F0 4040F140 40404040 0 NWSP00 1          Type-> Char
                                           00187DC8
00000010 40404040 40404040 40404040 40404040          00187DD8
00000020 40404040 40404040 40404040 40404040          00187DE8
00000030 40404040 40404040 40404040 40404040          00187DF8
OP2--> CA-AREA
00000000 00000000 00000000 00000000 05000018          Type-> Char
                                           05000060
00000010 009D7000 00186000 04F013A4 00000000          .....}.0..... 05000070
00000020 84D59EC6 8500018B 00000000 00000000          .N.F.....      05000080

1=HELP 2=PCK 3=EXIT 4=STEP 6=? 7=BWD 8=FWD 9=ZOOM 10=WINDOW 11=FLOW 12=CONT

```

In this case PF4 (STEP) was pressed and execution of program NWSP07 halts prior to the next statement (711) to be executed.

## Control Debugging and Execution using line commands

A debug session is made easy using LINE commands. You can control where execution of a program is halted, specify special halt conditions, and even alter the logic flow of the program using LINE commands. All of this can be done while viewing the source code of the program.

Halt points can be set in many ways, but the most convenient way is while viewing a module's source code. You can move around in the program's source by paging forward and backward, FINDing a string of characters, or going directly to a statement number.

Once the statement you want is located, simply move the cursor to anywhere in the statement number, key 'H' and press Enter. The halt point is set and when this statement is encountered during execution TRACK will halt the execution and produce a STOP Display screen.

In the same way, you can turn a Halt off. You find the statement, key 'R' (Reset) in the statement number, and press Enter.

```
=5.1 TRACK - Stop Display CICSUSER T08A 01/14/98 13.33.53
Command-----> -
Offset-----> Password---->

Stop Program--> NWSP07 Current Module-> NWSP07 At Terminal---> T08A
Phase Offset--> 001268 Module Offset--> 001248 Statement No.-> 000925
Stop Reason---> STEP REQUEST
Stmnt .....1.....2.....3.....4.....5.....6.....7..
000922 049200 GO TO 2000-NOTFND.
000923 049300
000924 049400 2000-LOOP-NWSFILE.
000925 049500 PERFORM 9020-READPREV-NWSFILE THRU 9020-EXIT.
000926 * DFHRESP(ENDFILE) = 20 INSERTED BY TRANSLATOR
H00927 049600 IF WS-RESP EQUAL 20
000928 049700 GO TO 2000-ENDFILE.
000929 049800 IF NWS-KEY EQUAL ALL 'g'
000930 049900 GO TO 2000-LOOP-NWSFILE.
R00931 050000 IF CA-LAST-ACCESS NOT EQUAL ZEROES
000932 050100 AND CA-LAST-ACCESS GREATER NWS-RELEASE
000933 050200 GO TO 2000-ENDFILE.
000934 050300 PERFORM 2900-SECURITY-CHECK THRU 2900-EXIT.
000935 050400 IF SECURITY-ERRORS

1=HELP 2=PCK 3=EXIT 4=STEP 6=? 7=BWD 8=FWD 9=ZOOM 10=WINDOW 11=FLOW 12=CONT
```

To set a Halt, key 'H' in the statement number and press Enter.

To Reset (turn a halt point off), key 'R' in the statement number.

The source of your program is displayed on your request using **TRACK** menu screens. It is also displayed during a debug session on a STOP screen. The above example shows a typical screen that was displayed due to a halt. It illustrates the setting and resetting of additional halt points.

Please note that at any halt point all **TRACK** debugging facilities are available. After each halt the programmer may terminate the transaction, produce a dump, continue execution normally, STEP through the program by statement or instruction, or continue from a different point in the program.

Besides PF key functions the programmer has available several keyable fields to control testing.

For instance, to halt at statement 921 only when field WS-RESP is not equal to 13, enter the following:

```
=7.2          TRACK - Halt Point Management  CICSUSER T08A 01/14/98 11.46.22
Command----->
Offset----->          Password----->

          Add New Halt Point
Terminal Id--> T08A          Task Id----->
Program Name--> NWSPO7      Module Name--> NWSPO7

Halt Offset--> s921          Halt Type----->
Times to Halt->
1st Halt Pass->
Pass Increment>
Halt Condition> ws-resp ne 13_

Storage Protection->

Date Access Action-> HALT

1=HELP 2=STOP 3=EXIT 4=STEP 6=? 11=FLOW 12=CONT
```

Each time program execution passed through statement number 921(Halt Offset —> S921), the contents of WS-RESP would be compared with a two-byte character value of 13. If they were not equal, a halt would occur.

```
=5.1          TRACK - Stop Display          CICSUSER T08A 01/14/98 12.10.09
Command-----> _
Offset----->          Password----->

Stop Program--> NWSPO7      Current Module-> NWSPO7      At Terminal--> T08A
Phase Offset--> 001254      Module Offset--> 001234      Statement No.-> 000921
Stop Reason--> HALT REQUEST - HALT ID = 3
  Stmt  ....+....1....+....2....+....3....+....4....+....5....+....6....+....7...
000918          GO TO 2000-ENDFILE DEPENDING ON DFHEIGDI.
000919 049000      PERFORM 9000-STARTBR-NWSFILE THRU 9000-EXIT.
000920          * DFHRESP(NOTFND) = 13          INSERTED BY TRANSLATOR
X00921 049100      IF WS-RESP EQUAL 13

-----
OP1--> WS-RESP
00000000 00000000          0          Type-> Binary          05000258

-----
000922 049200      GO TO 2000-NOTFND.
000923 049300
000924 049400 2000-LOOP-NWSFILE.
000925 049500      PERFORM 9020-READPREV-NWSFILE THRU 9020-EXIT.
000926          * DFHRESP(ENDFILE) = 20          INSERTED BY TRANSLATOR
000927 049600      IF WS-RESP EQUAL 20

1=HELP 2=PCK 3=EXIT 4=STEP 6=? 7=BWD 8=FWD 9=ZOOM 10=WINDOW 11=FLOW 12=CONT
```

Here, execution stops at statement 921 when WS-RESP is not equal to 13.

```
=5.1          TRACK - Stop Display          CICSUSER T08A 01/14/98 13.35.48
Command-----> _
Offset----->          Password----->

Stop Program--> NWSPO7      Current Module-> NWSPO7      At Terminal--> T08A
Phase Offset--> 0029A8      Module Offset--> 002988      Statement No.-> 001697
Stop Reason--> DATE ACCESS
  Stmt  ....+....1....+....2....+....3....+....4....+....5....+....6....+....7...
001694          * ABSTIME (WS-ABSTIME)
001695          * END-EXEC.
001696 115700      MOVE 'c          01161          ' TO DFHEIV0
001697          CALL 'DFHEI1' USING DFHEIV0 WS-ABSTIME.
001698
001699          *EXEC CICS FORMATTIME
001700          * ABSTIME (WS-ABSTIME)
001701          * YEAR (WS-ABSTIME-YEAR)
001702          * DAYOFMONTH (WS-ABSTIME-DOY)
001703          * MONTHOFYEAR (WS-ABSTIME-MOY)
001704          * TIME (WS-ABSTIME-TIME)
001705          * END-EXEC.
001706 116000      MOVE 'c          01164          ' TO DFHEIV0
001707          CALL 'DFHEI1' USING DFHEIV0 WS-ABSTIME DFHDUMMY DFHDUMMY

1=HELP 2=PCK 3=EXIT 4=STEP 6=? 7=BWD 8=FWD 9=ZOOM 10=WINDOW 11=FLOW 12=CONT
```

Execution was halted at statement 1697 when a system date was accessed. This was set via Date Access Action in option 8.2.

The Display/Debug facility provides access to all CICS and user tables, to program storage, and when used interactively to all areas associated with an active task. Modification to such areas can be made to correct errors.

Show and Modify Storage Areas  
within your Program by simply entering the field's dataname.

If the command field contained a field name such as CA-AREA, then the display would show the address and storage value for CA-AREA as displayed here.

```
=3.1 TRACK - Storage Area Display CICSUSER T08A 01/14/98 13.52.29
Command-----> -
Offset-----> Password---->

Display of--> D CA-AREA Type-> Char

00000000 F040D5E6 E2D7F0F7 4040F140 40404040 0 NWSP07 1 05000060
00000010 40404040 40404040 40404040 40404040 05000070
00000020 40404040 40404040 40404040 40404040 05000080
00000030 40404040 40404040 40404040 40404040 05000090
00000040 40404040 40404040 40404040 40404040 050000A0
00000050 40404040 40404040 4040E440 4040C3C9 U CI 050000B0
00000060 C3E2E4E2 C5D9D7E5 E2C5E3F0 F8C1E3C5 CSUSERPVSET08ATE 050000C0
00000070 E2E3F4F1 F040D5E6 E2D4E2E7 C7C3C65B ST410 NWSMSXGCF$ 050000D0
00000080 40D5E6E2 C6C9D3C5 407CC1C4 D4C9D540 NWSFILE @ADMIN 050000E0
00000090 40400007 D5D54040 40404040 4040D5E6 .NN NW 050000F0
000000A0 E2C1D5E6 E2E2D5E6 E2E4E840 40D4E2E7 SANWSSNWSUY MSX 05000100
000000B0 C7C3C65B 40404040 40404040 40404040 GCF$ 05000110
000000C0 40404040 40404040 40404040 40404040 05000120
000000D0 40404040 40404040 40404040 40404040 05000130
000000E0 D5E6E2D7 D5E6E2C3 D4C1C3F1 D4C1C3F2 NWSPNWSCMAC1MAC2 05000140
000000F0 D4C1C3D7 C8D3D740 40404040 40404040 MACPHLP 05000150

1=HELP 2=STOP 3=EXIT 4=STEP 5=RFIND 6=? 7=BWD 8=FWD 11=FLOW 12=CONT
```

The leftmost column displays the address relative to the data field displayed. The rightmost column shows the virtual storage address. The data in the middle of the display is the usual hexadecimal and character display of data for the field requested. In this case the data field CA-AREA is an 01 level field, so all data for the group is displayed.

When a program is halted prior to executing a statement the values of the variable names in the statement are displayed in a window.

In this example EIBCALEN has both its hex and decimal value displayed (01F4 and 500). It can be changed by keying over either one of the representations. When program execution is continued, processing will continue based on the new value.

```
=5.1 TRACK - Stop Display CICSUSER T08A 01/14/98 14.10.24
Command----->
Offset-----> Password---->

Stop Program--> NWSP07 Current Module-> NWSP07 At Terminal---> T08A
Phase Offset--> 0009DC Module Offset--> 0009BC Statement No.-> 000710
Stop Reason--> HALT REQUEST - HALT ID = 1
  Stmt .....1.....2.....3.....4.....5.....6.....7..
000707 029200
000708 029300 PROCEDURE DIVISION USING DFHEIBLK DFHCOMMAREA.
000709 029400 0000-INITIALIZATION-AND-CONTROL.
000710 029500 IF EIBCALEN GREATER ZERO

-----
DP1--> EIBCALEN Type-> Binary
00000000 01F4 500 001800E8
-----

000711 029600 MOVE DFHCOMMAREA TO CA-AREA
000712 029700 ELSE
000713 029800 GO TO 9999-END.
000714 029900
000715 030000 IF CA-PROGRAM-ID EQUAL 'P07'
000716 030100 GO TO 1000-MAIN-MENU.

1=HELP 2=PCK 3=EXIT 4=STEP 6=? 7=BWD 8=FWD 9=ZOOM 10=WINDOW 11=FLOW 12=CONT
```

There are other methods to change a data field's values.

In this example, the cursor was placed on WS-MAPNAME (line 1827) and PF9 was pressed. WS-MAPNAME can be altered (in either hex or character format) with this window.

```
=1.1 TRACK - STOP Program Source Display CICSUSER T08A 01/14/98 14.05.04
Command----->
Offset-----> Password---->
Phase Name--> NWSP07 Module Name-> NWSP07 Stmt-> 1820 Zone-> 1 120

Module was compiled on 01/14/98 at 11.12.42
 Stmt .....1.....2.....3.....4.....5.....6.....7..
001820
001821
001822 * DFHRESP(NORMAL) = 0 INSERTED BY TRANSLATOR
001823 123900 IF WS-RESP EQUAL 0
001824 * DFHRESP(MAPFAIL) = 36 INSERTED BY TRANSLATOR
001825 124000 OR WS-RESP EQUAL 36
001826 124100 GO TO 9340-EXIT.
001827 124200 MOVE WS-MAPNAME TO NWSE000-RESOURCE.
-----
Display of--> D=WS-MAPNAME Type-> Char
00000000 D5E6E2D4 C1D74040 NWSMAP 05000050
-----
001832 124700 MOVE +1 TO TRSP05-LENGTH.
001833 124800 CALL 'TRSP05' USING NWSM07-OPT TRSP05-LENGTH.
001834 124900 9341-EXIT. EXIT.

1=HELP 2=STOP 3=EXIT 4=STEP 5=RFIND 6=? 7=BWD 8=FWD 9=ZOOM 11=FLOW 12=CONT
```

On any TRACK display enter D and a data field's name in the command, NWSM07-OPT in this case.

```
=5.1 TRACK - Stop Display CICSUSER T08A 01/14/98 14.09.00
Command-----> d nwsM07-opt_ Password---->
Offset----->

Stop Program--> NWSP07 Current Module-> NWSP07 At Terminal---> T08A
Phase Offset--> 000BF8 Module Offset--> 000BD8 Statement No.-> 000774
Stop Reason---> STEP REQUEST
 Stmt .....1.....2.....3.....4.....5.....6.....7..
000771 035500 GO TO 1400-END-FUNCTION.
000772 035600* P-PREVIOUS
000773 035700 IF NWSM07-OPT EQUAL 'P'
000774 035800 GO TO 1410-PREVIOUS-FUNCTION.
000775 035900* N-NEXT
000776 036000 IF NWSM07-OPT EQUAL 'N'
000777 036100 GO TO 1420-NEXT-FUNCTION.
000778 036200* L-LATEST
000779 036300 IF NWSM07-OPT EQUAL 'L'
000780 036400 GO TO 1430-LATEST-FUNCTION.
000781 036500* O-OLDEST
000782 036600 IF NWSM07-OPT EQUAL 'O'
000783 036700 GO TO 1440-OLDEST-FUNCTION.
000784 036800* START DATE SPECIFIED?

1=HELP 2=PCK 3=EXIT 4=STEP 6=? 7=BWD 8=FWD 9=ZOOM 10=WINDOW 11=FLOW 12=CONT
```

Any part of the displayed storage may be modified, by positioning the cursor at the desired location in either the hex or character representation of the area, overtyping the existing value with the required one, and pressing the ENTER key.

```
=3.1 TRACK - Storage Area Display CICSUSER T08A 01/14/98 14.15.11
Command-----> -
Offset----->
Password---->
Display of--> D CA-PROGRAM-ID Type-> Char
00000000 D7F0F740 40 P07 05000065

1=HELP 2=STOP 3=EXIT 4=STEP 5=RFIND 6=? 7=BWD 8=FWD 11=FLOW 12=CONT
```

To change P07 to P77 either overtype the number 0 and change to 7, or hexadecimal F0 and change to F7. This screen would display.

```
=3.1 TRACK - Storage Area Display CICSUSER T08A 01/14/98 14.15.52
Command-----> -
Offset----->
Password---->
TRK060 - DATA CHANGES COMPLETED
Display of--> D CA-PROGRAM-ID Type-> Char
00000000 D7F7F740 40 P77 05000065

1=HELP 2=STOP 3=EXIT 4=STEP 5=RFIND 6=? 7=BWD 8=FWD 11=FLOW 12=CONT
```

Certain areas owned and maintained by CICS itself may be viewed and, in some cases, altered. To see the contents of the EIB, for example, we could have entered EIB on the command line.

```

=4.1          TRACK - CICS Area Display  CICSUSER T08A 01/14/98 14.16.42
Command-----> _
Offset----->      Password----->

Display of--> EIB

0000 EIBTIME  0141021C   14.10.21   003F EIBATT   00
0004 EIBDATE  0100001F   00/001    0040 EIBEOC   00
0008 EIBTRNID                NWSS      0041 EIBFMH   00
000C EIBTASKN 0047901C   47901     0042 EIBCOMPL 00      DO RECEIVE
0010 EIBTRMID                T08A      0043 EIBSIG   00
0016 EIBCPOSN 0726      R23 C71   0044 EIBCONF  00
0018 EIBCALEN 01F4      500       0045 EIBERR   00
001A EIBAID   7D        ENTER     0046 EIBERRCD 00000000
001B EIBFN    0E04      XCTL
001D EIBRCODE 000000000000  NORMAL
0023 EIBDS
002B EIBREQID
0033 EIBSRCE                NWSP07
003B EIBNODAT 00
003C EIBFREE  00
003D EIBRECV  00

004A EIBSYNRB 00
004B EIBNODAT 00
004C EIBRESP  00000000
0050 EIBRESP2 00000000
0054 EIBRLDBK 00

1=HELP 2=STOP 3=EXIT 4=STEP 6=? 11=FLOW 12=CONT

```

**If EIB was entered in the command field of the previous halt screen and Enter pressed, we could display the contents of the user's Exec Interface Block as shown here.**

The contents of such areas as the CSA, TCA, CWA, TWA, TCTTE, TUA, and others may be examined in a similar manner.

## Extended Halt Points

Extended halt points are similar to standard halts, but when setting extended halt points the user may specify that the halt is only to take effect if certain conditions are met. For example, a halt could be requested to take place at program statement number 999 when data field WKTERM is equal to T05A. It is possible to make the halt point even more selective by also specifying:

a limit to the number of times it will be honored;

how many passes through the halt point must occur, with any condition met, before the halt takes effect;

how many passes through the halt point must occur, with any condition met, between halts;

any of the relation identifiers, equals, not equals, greater than, greater than or equal, less than, less than or equal can be used.

## Alteration of Program Flow

If the programmer decides to continue execution of the program, the continuation can be made at a point other than that at which the program was last halted. The user's display, if saved, may or may not be restored depending upon the continuation option selected. When the program is terminated, a dump may or may not be requested as the programmer wishes.

## Files and Temporary Storage Management

Track offers on-line access to all CICS files which may be browsed, updated, added to, or deleted from. Temporary storage queues may also be created, updated, or deleted. These facilities are available regardless of whether you are using **TRACK** to test a program.

```

=F                                     TRACK - File Management      CICSUSER T08G 02/19/'
Action----> WEXT                      Filename-> MSXGCF$
Key/Addr-> C'001B'NWS

Offset----> 0                          Reclen----> FA                      Password:-
Filetype-> VSAM KSDS                    RecForm----> VB                      Key Posn:-
                                           Keylen----> 30

0000    001B05E6  E2404040  40404040  40404040  ..NWS
0010    40404040  40404040  40404040  40404040
0020    40404040  40404040  40404040  40404040
0030    0007D5D5  40404040  40404040  D5E6E2C1
0040    D5E6E2E2  D5E6E2E4  7CC1C4D4  C9D54040  ..NN      NI
0050    40E8C6F0  C1F0C3F0  F0F2C5F7  F4F3F5F8  NWSSNWSU@ADMII
0060    C5F44040  40404040  40404040  40404040  YF0A0C002E74:
0070    40404040  40404040  40404040  40404040  E4
0080    40404040  40404040  40404040  40404040
0090    40404040  40404040  40404040  40404040
00A0    40404040  40404040  40404040  40404040
00B0    4040D5E6  E2C34040  40404040  40404040  NWS
00C0    40404040  40404040  40404040  40404040
00D0    40404040  40404040  40404040  40404040
00E0    40404040  40404040  40404040  40404040
00F0    40404040  40404040  40404040  40404040
1=HELP 3=EXIT 5=PREV 7=BWD 8=FWD
  
```

Here is an example of a display of file 'MSXGCF\$'.

```

=Q                                     TRACK - TSQ Management      CICSUSER T08G 02/19/'
Action-----> WEXT
Queue Name--> QEDBT08G                  Item No--> 1
Offset-----> 0                          Length----> 7FFA                      Password:-

0000    00020000  00010002  00030004  00050006  .....
0010    00070008  00000000  0009000A  000B000C  .....
0020    000D000E  000F0010  00000000  00110012  .....
0030    00130014  00150016  00170018  00000000  .....
0040    0019001A  001B001C  001D001E  001F0020  .....
0050    00000000  00210022  00230024  00250026  .....
0060    00270028  00000000  0029002A  002B002C  .....
0070    002D002E  002F0030  00000000  00310032  .....
0080    00330034  00350036  00370038  00000000  .....
0090    0039003A  003B003C  003D0000  00000000  .....
00A0    00000000  00000000  00000000  00000000  .....
00B0    00000000  00000000  00000000  00000000  .....
00C0    00000000  00000000  00000000  00000000  .....
00D0    00000000  00000000  00000000  00000000  .....
00E0    00000000  00000000  00000000  00000000  .....
00F0    00000000  00000000  00000000  00000000  .....
1=HELP 3=EXIT 5=PREV 7=BWD 8=FWD
  
```

This is a sample of a Temporary Storage Queue display for queue QEDBT08G.

## Security

Access to all **TRACK** facilities may be restricted using the system's own internal security. User profiles may be defined to limit the functions available to individuals and resource profiles can be set up to protect files and temporary storage queues. Details of all changes made by **TRACK** users to programs, storage areas, files, and temporary storage are written to the transient data destination CSSL.

## System Testing

When programs are submitted for integration testing, errors can arise due to the interaction of programs with each other. The problems that occur at this stage of testing are often difficult to debug and can cause CICS to crash. By using **TRACK** to monitor all tasks activated during such system testing, problems can also be investigated and corrected interactively, again allowing the testing process to proceed further than it otherwise would and potentially improving the CICS system availability.

## Initial Production Running

Even the most exhaustive testing may not eliminate all errors in a new application system. Thus when a new application is introduced in the production CICS system, the **TRACK** system can be used to protect the integrity of the system by monitoring the new and/or changed programs for a period of time.

## Separate Terminal Debugging

Debugging activity can be specified to occur at a separate terminal other than the one at which the monitored program is being executed. This facilitates the centralized control of errors in a production system and enables the debugging of programs which are executed at non-3270 type terminals. In this way transactions which are not terminal-attached can also be monitored.

## Supported Environments

*VSE:* all releases

*MVS:* all releases

*CICS:* 1.7-4.1

*Languages:* Cobol, Cobol II, COBOL LE/370, PL/1, Assembler

## Free Trial

**TRACK** is easy to install and use. There are no modifications to CICS required, except the standard table entries. **TRACK** appears to the system as a task under CICS.

Try **TRACK** in your own installation for 30 days FREE. Find out first hand the many benefits of **TRACK** that other users throughout the U.S. and Europe have already realized.

**TRACK** is a proprietary product of BITS Software, Ltd., of England. MacKinney Systems is a marketing agent for BITS Software and handles sales and technical support.