z/TPF TRAG Driver

User's Guide

This page intentionally left blank.

ZTEST TRAG

The purpose of the TRAG driver is to verify that an ECB will use its first tracegroup buffer (the IBM_DEFT buffer) after exceeding its number of allocated tracegroup buffers. In this document, the term *tracegroup buffer* refers to both the C function trace buffer and the macro trace buffer for a tracegroup. For example, if an ECB has 5 tracegroup buffers (1 for the IBM_DEFT tracegroup and 4 for other tracegroups), the fifth new tracegroup encountered will have its function calls/macros stored in the IBM_DEFT tracegroup buffer.

The TRAG driver tests all combinations for the transition to the final tracegroup:

- CSO to CSO
- CSO to BSO
- BSO to BSO
- BSO to CSO

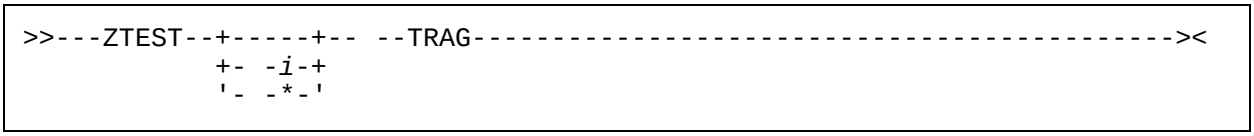
A new ECB is created for each of these cases. Each ECB changes the tracegroup for the program it is executing and then restarts that program, or it changes the tracegroup for the other type of program (BSO or CSO) and starts executing that program. When the number of tracegroups has exceeded the number of buffers by 1, a check is performed to verify that the current tracegroup buffer being used is actually the first tracegroup buffer (the IBM_DEFT buffer).

The ECB behavior is determined by two parameters that are passed in the EBW area. The first parameter is the number of times the ECB must change its tracegroup and restart (initially equal to the number of tracegroup buffers in the ECB), and the second parameter is the number of restarts left before the ECB should jump to the other type of program (CSO or BSO). For the second parameter, -1 represents no switch to the other type of program.

Requirements and restrictions

None.

Format



i

indicates the specific I-stream in which the driver will be run. If *i* is not specified, the test case(s) will be executed on the I-stream on which the command is entered.

****

specifies the driver will be invoked on all currently defined and available I-streams.

Source code information

The TRAG driver consists of the following program segments:

Header Files

Header File	Description
trag.h	Contains constants such as the names of the CSO and BSO test programs and the value for the NO_SWITCH parameter. It also contains the iecb_tr_buffer_list structure definition taken from collatetrace.c and the declaration for the alter_tracegroup() function.

BSOs

Module	Makefile	Segment	Description
QGKA	qgka.mak	qgka.asm	The BSO program that will perform the tests. See *Additional information* for a description for the series of steps taken during a test.

CSOs

Module	Makefile	Segment	Description
QGKC	qgkc.mak	qgkc.c	The CSO program that will perform the tests. See *Additional information* for a description for the series of steps taken during a test.
QTRG	qtrg.mak	qtrg.c	The entry point for the driver. This code uses wtopc() to print which test case is about to be run and prepares the EBW parameters that are needed for that case. It then executes a tpf_fork() to create a new ECB to perform the test. The waitpid() function is used to pause execution until the child ECB has finished its test.

Additional information

- Upon starting, it examines the two parameters passed to it through the EBW area. If the second parameter (restarts_before_switch) is zero, the tracegroup for the BSO test program is altered, both EBW parameters are decremented (effectively setting restarts_left to zero and resarts_before_switch to -1), and then an entdc() is used to switch over to the BSO test program.
- Otherwise, if the first parameter (restarts_left) is greater than zero, the tracegroup for the CSO test program is altered, restarts_left is decremented, restarts_before_switch is decremented if necessary (if it is not -1), and an entdc() is used to restart execution of the CSO test program.
- If neither of the above conditions are true, then there is no switch and no restarts left. Page 3 for the ECB is accessed to obtain pointers to the current C function trace buffer (ce3curch) and the current macro trace buffer (ce3curmh). Page 2 for the ECB is accessed to examine the buffer list (ce2trlist), which contains pointers to the starts of the C function trace buffers and the macro trace buffers. These pointers should be equal to the current pointers

obtained from Page 3, indicating that the ECB is now using the first C function trace buffer and the first macro trace buffer (both of which are for the IBM_DEFT tracegroup). A wtopc() displays whether the correct or incorrect buffers are being used.

- File qgkc.c also contains the implementation for the alter_tracegroup() function, which is used by both the CSO and BSO test programs. This function accesses the PAT for the designated program and changes its tracegroup to "TGi", where i is the index value of the new tracegroup (1 greater than the last tracegroup index).

Examples

The following example shows a successful run of the TRAG driver:

```
User:   ZTEST TRAG

System: QTRG0001I 16.39.39 TESTING CSO TO CSO+
        QGKC0001I 16.39.39 CORRECT BUFFERS USED+
        QTRG0001I 16.39.39 TESTING CSO TO BSO+
        QGKA0001I 16.39.39 CORRECT BUFFERS USED+
        QTRG0001I 16.39.39 TESTING BSO TO BSO+
        QGKA0001I 16.39.39 CORRECT BUFFERS USED+
        QTRG0001I 16.39.39 TESTING BSO TO CSO+
        QGKC0001I 16.39.39 CORRECT BUFFERS USED+
```

Messages

Below is a list of the TRAG driver messages:

QGKA0001I CORRECT BUFFERS USED

Explanation: Both the current C function trace buffer and the current macro trace buffer are the first buffers.

System action: None.

User response: None.

QGKA0001E ERROR: INCORRECT BUFFERS USED

Explanation: The final check for a case revealed that the current tracegroup buffer does not match the IBM_DEFT tracegroup buffer.

System action: None.

User response: None.

QGKC0000E ERROR LOCATING PAT FOR PROGRAM *program_name*

Explanation: In order to modify the tracegroup for a program, the program attribute table (PAT) is accessed for that program using the progc() function. The PAT cannot be located for *program_name*.

System action: The driver exits.

User response: None.

QGKC0001I CORRECT BUFFERS USED

Explanation: Both the current C function trace buffer and the current macro trace buffer are the first buffers in the list since the ECB is now using the IBM_DEFT buffers.

System action: None.

User response: None.

QGKC0001E ERROR: INCORRECT BUFFERS USED

Explanation: The final check for a case revealed that the current tracegroup buffer does not match the IBM_DEFT tracegroup buffer.

System action: None.

User response: None.

QTRG0001I TESTING *module* TO *module*

Where:
module is either CSO or BSO.

Explanation: One of the four variations between CSO and BSO is being tested for the transition to the final tracegroup (CSO to CSO, CSO to BSO, BSO to BSO, and BSO to CSO).

System action: None.

User response: None.

References

For more information about reading syntax diagrams, also referred to as railroad diagrams, see *Accessibility information* in the TPF Product Information Center.