

CEEGTST—Get heap storage

CEEGTST—Get heap storage

CEEGTST gets storage from a heap whose ID you specify. It is used to acquire both large and small blocks of storage. CEEGTST always allocates storage that is addressable by the caller. Therefore, if the caller is in 24-bit addressing mode, or if HEAP(„BELOW) is in effect, the storage returned is always below the 16M line. Above-the-line storage is returned only if the caller is in 31-bit addressing mode and HEAP(„ANY) is in effect.

All requests for storage are conditional. If storage is not available, the feedback code (fc) is set and returned to you, but the thread does not abend. When storage is not available, the appropriate action in the member environment should be taken. One option is to use the CEESGL callable service to signal the Language Environment condition handler with the returned feedback code.

Storage obtained by CEEGTST can be freed by a call to CEEFRST or CEEDSHP. You can also free storage by using a language intrinsic function. If storage is not explicitly freed, it is freed automatically at termination.

If you have specified a heap_alloc_value in the STORAGE runtime option, all storage allocated by CEEGTST is initialized to heap_alloc_value. Otherwise, it is left uninitialized.

If the value specified in the size parameter of CEEGTST is greater than the size of an increment (as specified in the HEAP runtime option), all of the requested storage (rounded up to the nearest doubleword) is allocated in a single system-level call.

Heap storage is acquired by a system-level get storage call in increments of init_size and incr_size bytes as specified by the HEAP runtime option, or in the CEECRHP callable service. If the increment size is chosen appropriately, only a few of the calls to CEEGTST result in a system call. The storage report generated when the RPTSTG runtime option is specified shows the number of system-level get storage calls required. This helps you tune the init_size and incr_size fields in order to minimize calls to the operating system.

►► CEEGTST (— heap_id — , — size — , — address — , — fc —)

Read syntax diagramSkip visual syntax diagram

- heap_id (input)** A fullword binary signed integer. heap_id is a token denoting the heap in which the storage is allocated. A heap_id of 0 allocates storage from the initial heap (or user heap). Any other heap_id must be a value obtained from the CEECRHP callable service. If the heap_id you specify is invalid, no storage is allocated. CEEGTST terminates with a non-CEE000 symbolic feedback code and the value of the address parameter is undefined.
- size (input)** A fullword binary signed integer. size represents the amount of storage allocated, in bytes. If the specified amount of storage cannot be obtained, no storage is allocated, CEEGTST terminates with a non-CEE000 symbolic feedback code, and the value of the address parameter is undefined.
- address (output)** A fullword address pointer. address is the main storage address of the first byte of allocated storage. If storage cannot be obtained, address remains undefined. Storage is always allocated on a doubleword boundary. This parameter contains an address that is returned as output to CEECZST.
- fc (output)** A 12-byte feedback code, optional in some languages, that indicates the result of this service. If you choose to omit this parameter, refer to Invoking callable services for the appropriate syntax to indicate that the feedback code was omitted.

The following symbolic conditions can result from this service:

Code	Severity	Message number	Message text
CEE000	0	—	The service completed successfully.
CEE0P2	4	0802	Heap storage control information was damaged.
CEE0P3	3	0803	The heap identifier in a get storage request or a discard heap request was unrecognized.
CEE0P8	3	0808	Storage size in a get storage request (CEEGTST) or a reallocate request (CEECZST) was not a positive number.
CEE0PD	3	0813	Insufficient storage was available to satisfy a get storage (CEECZST) request.

Usage notes

- COBOL considerations— If you want to use the CEEGTST callable service in a 24-bit addressing mode COBOL program to request 24-bit heap storage,

you can make a static call to CEEGTST without any additional changes. If you want to call it dynamically (either by using the COBOL DYNAM compiler option, or using the "CALL identifier" statement, where identifier is a variable that holds the name of the program you want to call), you must also specify HEAP(„BELOW„,) as a runtime option.

This runtime specification affects all programs in the enclave using the user heap, not just the 24-bit addressing mode program. If this is undesirable, one alternative is to use the CEECRHP callable service to create an additional heap, specifying for the options parameter a value that will create the heap BELOW. The 24-bit addressing mode COBOL program can then obtain storage from this additional heap using the CEEGTST callable service.

- PL/I considerations—Storage allocated within PL/I AREAs is managed by PL/I. Therefore, only PL/I language functions can allocate and free storage within a PL/I area.
- Based upon the layout of a PL/I structure, PL/I might adjust the starting byte of the PL/I structure to a non-doubleword aligned byte. The difference between the doubleword boundary and the first byte of such a structure is known as the hang. Because Language Environment callable storage services do not adjust the starting byte, you must be careful using callable services to allocate storage for PL/I structures. Use either the PL/I ALLOCATE statement or fully defined structures and aggregates to avoid this problem.
- CICS® considerations—In a CICS environment, size should not exceed 1024M (1 gigabyte or X'40000000') when running in AMODE ANY, and 65,504 bytes when running in 24-bit addressing mode. These CICS restrictions are subject to change from one release of CICS to another. Portable applications should respect current CICS limitations.
- z/OS UNIX considerations—CEEGTST applies to the enclave. Any thread can free the allocated storage.

For more information

- See HEAP for further information about the HEAP runtime option.
- For more information about the CEESGL callable service, see CEESGL—Signal a condition.
- For more information about the CEEFRST callable service, see CEEFRST—Free heap storage.
- For more information about the CEEDSHP callable service, see CEEDSHP—Discard heap.
- For more information about the STORAGE runtime option, see STORAGE.

- For more information about the CEECRHP callable service, see CEECRHP—Create new additional heap.
- For more information about the RPTSTG runtime option, see RPTSTG.
- For more information about the CEECRHP callable service, see CEECRHP—Create new additional heap.
- For more information about the CEECZST callable service, see CEECZST—Reallocate (change size of) storage.

Examples

1. Following is an example of CEEGTST called by C/C++.

```

/*Module/File Name: EDCGTST  */

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <leawi.h>
#include <ceedcct.h>

int main(void) {

    _INT4 heapid, size;
    _POINTER address;
    _FEEDBACK fc;
    /* .
     .
     . */
    heapid = 0;    /* get storage from initial heap */
    size = 4000;   /* number of bytes of heap storage */

    /* obtain the storage using CEEGTST */
    CEEGTST(&heapid,&size,&address,&fc);

    /* check the first 4 bytes of the feedback token */
    /* (0 if successful) */
    if ( _FBCHECK ( fc , CEE000 ) != 0 ) {
        printf("CEEGTST failed with message number %d\n",
               fc.tok_msgno);
        exit(99);
    }
    /* .
     .
     . */
    /* free the storage that was previously obtained */
    /* using CEEGTST */

```

```

CEEFRST(&address,&fc);

/* check the first 4 bytes of the feedback token */
/* (0 if successful) */
if ( _FBCHECK ( fc , CEE000 ) != 0 ) {
    printf("CEEFRST failed with message number %d\n",
        fc.tok_msgno);
    exit(99);
}
/* .
.
. */
}

```

2. Following is an example of CEEGTST called by COBOL.

```

CBL LIB,QUOTE
*Module/File Name: IGZTGTST
*****
**
** IGZTGTST - Call CEEGTST to get heap storage
**
** In this example, a call is made to CEEGTST to
** obtain 4000 bytes of storage from the initial
** heap (HEAPID=0).
**
*****
IDENTIFICATION DIVISION.
PROGRAM-ID. IGZTGTST.
DATA DIVISION.
WORKING-STORAGE SECTION.
01  HEAPID                PIC S9(9) BINARY.
01  STGSIZE                PIC S9(9) BINARY.
01  ADDRSS                POINTER.
01  FC.
02  Condition-Token-Value.
COPY CEEIGZCT.
03  Case-1-Condition-ID.
04  Severity              PIC S9(4) BINARY.
04  Msg-No                PIC S9(4) BINARY.
03  Case-2-Condition-ID
    REDEFINES Case-1-Condition-ID.
04  Class-Code            PIC S9(4) BINARY.
04  Cause-Code            PIC S9(4) BINARY.
03  Case-Sev-Ctl          PIC X.
03  Facility-ID           PIC XXX.
02  I-S-Info              PIC S9(9) BINARY.

```

```

PROCEDURE DIVISION.
*****
** Specify 0 to get storage from initial heap. **
** Specify 4000 to get 4000 bytes of storage. **
** Call CEEGTST to obtain storage.           **
*****
PARA-CBLGTST.
    MOVE 0 TO HEAPID.
    MOVE 4000 TO STGSIZE.

    CALL "CEEGTST" USING HEAPID, STGSIZE,
                        ADDRSS, FC.
    IF CEE000 of FC THEN
        DISPLAY "Obtained " STGSIZE " bytes of"
            " storage at location " ADDRSS
            " from heap number " HEAPID
    ELSE
        DISPLAY "CEEGTST failed with msg "
            Msg-No of FC UPON CONSOLE
        STOP RUN
    END-IF.

GOBACK.

```

3. Following is an example of CEEGTST called by PL/I.

```

*PROCESS MACRO;
/* Module/File Name: IBMGTST */
/*****/
/** */
/** Function: CEEGTST - Get Heap Storage */
/** */
/** In this example, a call is made to CEEGTST to */
/** request 4000 bytes of storage from the */
/** initial heap. */
/** */
/*****/
PLIGTST: PROC OPTIONS(MAIN);

    %INCLUDE CEEIBMAW;
    %INCLUDE CEEIBMCT;

    DCL HEAPID REAL FIXED BINARY(31,0);
    DCL STGSIZE REAL FIXED BINARY(31,0);
    DCL ADDRSS POINTER;
    DCL 01 FC, /* Feedback token */
        03 MsgSev REAL FIXED BINARY(15,0),

```

```

03 MsgNo      REAL FIXED BINARY(15,0),
03 Flags,
    05 Case      BIT(2),
    05 Severity  BIT(3),
    05 Control   BIT(3),
03 FacID      CHAR(3),          /* Facility ID */
03 ISI        /* Instance-Specific Information */
                REAL FIXED BINARY(31,0);

HEAPID = 0; /* get storage from the initial heap */
STGSIZE = 4000; /* get 4000 bytes of storage */

/* Call CEEGTST to obtain the storage */
CALL CEEGTST ( HEAPID, STGSIZE, ADDRSS, FC );
IF FBCHECK( FC, CEE000) THEN DO;
    PUT SKIP LIST( 'Obtained ' || STGSIZE
        || ' bytes of storage at location '
        || DECIMAL( UNSPEC( ADDRSS1 ) )
        || ' from heap ' || HEAPID );
    END;
ELSE DO;
    DISPLAY( 'CEEGTST failed with msg '
        || FC.MsgNo );
    STOP;
    END;

END PLIGTST;

```

Parent topic: Callable services