CAL16$ - Execute 16-bit Program from 32-bit

Although a 32-bit program can simply EXEC another 32-bit program a special technique must be used by a 32-bit program to execute a 16-bit program. The CAL16$ routine is one of a family of sub-routines that provide this functionality.

1. Invocation

To execute a 16-bit program from 32-bit code:

        CALL CAL16$ USING prog_name

where prog_name is a PIC X(8) field, or literal, that contains the name of the 16-bit program.

2. STOP Codes and Exception Conditions

No STOP codes are generated by CAL16$.

No EXIT codes are returned by CAL16$.

3. Programming Notes

CAL16$ is only available in GSM V8.1k, or later.

Note that the 16-bit program name must be exactly 8 characters (i.e. trailing SPACES must be added if the program name is less than 8 characters). Control will return to the instruction after the CALL CAL16$ when the 16-bit terminates with a STOP RUN statement. The 16-bit program will **not** return to the 32-bit program if it suffers a STOP code, EXIT code or any other exception.

All three sub-routines in the CAL16$, EX16C$, EX16S$ family save the screen image before executing the 16-program; and restore the screen image after the 16-bit program terminates. In addition, the CAL16$ and EX16C$ routines display a System Request On escape sequence before executing the 16-program; and display a System Request Off escape sequence after the 16-bit program terminates. The EX16S$ routine displays a Speedbase Mode Off escape sequence before executing the 16-program; and displays a Speedbase Mode On escape sequence after the 16-bit program terminates. Furthermore, the CAL16$ routine resets the $$INDE System Variable after the 16-bit program terminates.

4. Examples

[EXAMPLE REQUIRED]

5. Copy-Books

None.

6. See Also

EX16C$ Execute 16-bit program from 32-bit
EX16S$ Execute 16-bit program from 32-bit
BREXEC   Execute 16-bit Reporter report