CUSTX$ - Customise 32-bit Program File

The CUSTX$ sub-routine attempts to emulate the 16-bit CUST$ sub-routine as documented in the System Subroutines Manual. Note that the entry-point CUST$ is **not** available for 32-bit programs.

There are some important difference between 16-bit CUST$ and 32-bit CUSTX$ which are described below.

1. Invocation

To customise a section of a 32-bit program code:

```
CALL CUSTX$ USING addr leng
```

where addr is the address of the first section of the 32-bit program to customise; and leng is the length of the section.

2. STOP Codes and Exception Conditions

The following STOP codes may be generated by CUSTX$:

STOP code	Description
3502	It was not possible to open or write the last 32-bit program file loaded.
3503	The last program file loaded was not a 32-bit program.

No EXIT codes are returned by CUSTX$.

3. Programming Notes

CUSTX$ is only available with GSM V8.11, or later.

The 32-bit CUSTX$ routine is functionally very different from 16-bit CUST$. The 16-bit CUST$ routine does not require any parameters and writes back the **entire** 16-bit program area to the last 16-bit program loaded. The 32-bit CUSTX$ routine requires both an address and a length to specify the **section** of the 32-bit memory page that is written back to the same offset in the 32-bit program file.

Normally the start address of the program section to be customized will be in the DATA DIVISION (self-modifying code is **not** recommended) and can be a single data item, a group item or a number of consecutive data items. The leng parameter specifies the length of the block that is written back to the program file.

Multiple calls to CUSTX$ using different address/length parameters can be executed to customise several sections of a DATA DIVISION.

The CUSTX$ routine writes back a section of 32-bit memory to the last 32-bit program loaded (excluding any Dynamic Load Modules that are automatically loaded to satisfy external symbols required by the Program Loader). Thus, the following coding technique is one example that will produce unpredictable results, typically program file corruption:

```
EXEC "PROGRAM2"
CALL CUSTX$ USING X-AREA Z-LENG
```

4. Examples
[EXAMPLE REQUIRED]

5. Copy-Books
None.

6. See Also
None.