

TO: MSPM Distribution
FROM: N. Adleman
SUBJ: BN.7.08
DATE: 09/29/67

This revision of BN.7.08 contains a minor revision in the contents of the first word of data segments grown by datmk_. Some additional error codes have also been added.

Published: 09/29/67
 (Supersedes: BN.7.08, 04/21/67)

Identification

Data segment grower
 datmk_
 N. AdTeman, D.H. Johnson

Purpose

This section describes the datmk_ procedure which is used to create data segments when they are required during the execution of a process. In particular, datmk_ is used in the implementation of the PL/I external static storage feature. datmk_ is called by the linker as a result of an out-reference in a linkage section when the "trap before link" option (MSPM BD.7.01) has been requested.

By the time the faulting instruction that accesses the external symbol has finished executing, the data segment has been created or grown and, possibly, initialized, and the user's instruction has had its desired effect.

Usage

Use of datmk_ is specified in EPLBSA by:

	segref	datmk_,datmk_
	segref	segment,symbol(datmk_(arglist))
arglist	...	
	dec	size
	dec	initialswitch
	arg	initializer

Here segment and symbol are the names of a segment and an in-reference in that segment's linkage section. At execution time, the first reference to symbol, e.g. the instruction

```
eapbp      symbol
```

causes a trap to the linker, which in turn calls datmk_. If segment is not active in the process, datmk_ creates it, including its linkage section. Then, if symbol is not listed as an in-reference in segment's linkage section, datmk_ grows segment by size words and creates the in-reference pointing to the newly-grown storage.

If initialswitch is non-zero, datmk_ fills in the faulting link pair and calls the user's initializing procedure located at initializer. This call has the form of a call to a PL/I internal procedure (see BP. 3.00 for details) with no arguments. Since this call does not go through the linkage section, if the initializing routine uses the base pair $lb \leftarrow lp$ it must obtain the proper values itself. Assuming that $lb \leftarrow lp$ is properly set, however, the initializing routine may freely refer to symbol.

The linker calls datmk_ as follows:

```
call datmk_ (argpointer,panelpointer);
```

where argpointer is a pointer to the user's argument list as specified in the segref pseudo-op above, and panelpointer is a pointer to stored machine conditions as follows:

```
words    1-8   contain the base address registers
          9-16  contain the arithmetic registers
          17-23 contain the SCU information
```

Method

When datmk_ is called, it performs the following steps:

1. References to segment and symbol are established by building pointers to the character strings in the linkage section of the process that called datmk_.
2. With the "trap before definition" allowed, the generate_ptr procedure (MSPM BY.13.02) is called to determine the status of segment and symbol.
3. If the symbol is already defined, datmk_ returns control directly to its caller, with no further action. If the symbol is not defined, and the segment is known to the calling process, control continues at step 4. Otherwise, the data segment and its associated linkage segment are created by the setnamestatus procedure which sets the Segment Name Table and retrieves the segment pointers. The new data segment is created so that any user requiring a segment named segment has access (global access). The first word of the newly created data segment is set to two. (Note that the first word of all data segments created by datmk_ is reserved to contain a value which represents the current size, in words, of the data segment.)

datmk_ then guarantees that the value of the first word will always be even and non-negative before a data segment is grown. The header for the new linkage segment is initialized.

4. The link_change\$make definition procedure is called to insert a definition for symbol in the linkage segment. The contents of the first word of segment is used as the value argument to the link_change\$make_ definition procedure.
5. The size argument to datmk_ is verified to be even, non-zero and non-negative. The first word of segment is then incremented by the value of size. In effect, segment is "grown" by size words.
6. If there was only one argument to datmk_, (i.e. initialswitch is zero), control returns to the calling process. Otherwise, to force a link to the data segment (fill-in the faulting pair), the link_fault\$force procedure is called with the "trap before link" option ignored. Then, the caller's initializing routine, as specified by the initializer argument, is called to initialize the data segment. This is done by accessing the sb←sp information in the machine conditions that are passed to datmk_. When control is returned to datmk_ by the user's initializing routine, datmk_ returns complete control to the calling process.

Errors

The datmk_ procedure uses the standard error handling mechanism (MSPM BY.11) to report all abnormalities that may be encountered. The condition "datmk_err" is signalled for the following errors:

<u>Error No.</u>	<u>Meaning</u>
1	error determining if the segment is known to the process that invoked datmk_
2	an invalid value of "class" was returned by generate_ptr
3	error attempting to create the text segment

<u>Error No.</u>	<u>Meaning</u>
4	error attempting to create a linkage segment
5	datmk_ has been requested to grow a data segment by a negative or zero number of words
6	the current size (in words) of a previously created data segment has become either negative or zero