DSFMT$ - Convert Data to Displayable Format

The DSFMT$ routine is used to convert a data area to a displayable format.

1. Invocation

To convert a block of data to a displayable format code:

```
CALL DSFMT$ USING prog symbol input output
```

where *prog* is a PIC X(8) program name (i.e. page name), *symbol* is a PIC X(8) symbol name; *input* is a PIC X(50) field containing the data in original format and *output* is a PIC X(50) field into which the formatted data will be returned in the display format appropriate for the type of field.

2. STOP Codes and Exception Conditions

No STOP codes are generated by DSFMT$.

The following exception conditions may be returned by DSFMT$.

EXIT code	$$COND	Description
10702	2	The symbol cannot be found in the specified symbol table
10703	3	The field is not one of character, display numeric, computational, pointer or date
10704	4	The input field value is incompatible with the format of the symbol (e.g. it would overflow, invalid date etc.)

3. Programming Notes

Both the Program Name (page name) and the Symbol Name must be supplied. DSFMT$ searches the in-memory Symbol Table to locate the symbol. Thus, DSFMT$ can only be used to return the address of symbols that are referenced in the compilation.

For character fields (e.g. PIC X(10)), the appropriate number of characters for that field will be simply moved to the output field. Trailing SPACES will **not** be written to the output field.

Display numeric fields will be validated and simply moved to the output field.

Computational fields will be converted to the "best fit" display numeric format in the output field.

Date fields will be converted to long date display formats in the output field.

Pointer fields will be converted to hexadecimal format in the output field.

4. Examples

[EXAMPLE REQUIRED]

5. Copy-Books

None.

6. See Also

SIZOF$ Return size of data item in bytes
ADDOF$ Return address of symbol