



## 1 SUMMARY

This subroutine allows **several versions of a program to be held in a single non-executable master version**. Execution of this subroutine extracts a particular version. It is intended primarily for Fortran programs but can also be applied to programs in other languages.

**ATTRIBUTES** — **Version:** 1.0.0. **Types:** OE08A **Original date:** July 1982. **Origin:** S.Marlow and J.K.Reid, Harwell.

## 2 HOW TO USE THE PACKAGE

### 2.1 Form of the master version and the effect of the subroutine

The master version contains ordinary lines, alternative lines and control lines. Alternative lines contain a version code character in column 1. This may be any alphabetic character other than C. Control lines contain a special character in columns 1 and 2. The default choice is % and we will assume that this is in use during our description here.

A simple example of a master file containing a single precision and a double precision version is as follows

```
%%BEGIN
%%VERSION SINGLE=S
%%VERSION DOUBLE=D
S    REAL A,B
D    DOUBLE PRECISION A,B
      DO 20 I=1,1000
      READ(5,10)A
      10 FORMAT(F16.6)
S    B=ALOG(A)
D    B=DLOG(A)
      WRITE(6,15)B
S    15 FORMAT(1PE12.4)
D    15 FORMAT(1PD20.12)
      20 CONTINUE
      STOP
      END
%%END
```

When the version code of an alternative line is active, it is converted into an ordinary line, usually by replacing column 1 by a space. However

- (i) if column 2 contains the character then columns 3 to 72 are shifted left by two positions and columns 71,72 are set to spaces (e.g. to construct a Fortran comment or a line of JCL) and
- (ii) if columns 2 to 6 all hold digits then columns 2 to 6 are shifted left by one position and column 6 is set to a space (to bring a 5-digit label forward).

Control lines have one of the following forms

- (a)

This results in all settings of earlier control lines being cancelled. Processing continues exactly as when the first line is read.

(b)

If there is a character and it is an active version code or if there is no character, then all lines between these two lines are copied unchanged, including those whose format is that of an alternative or control line.

(c)

This changes the special character

(d)

This defines alphabetic characters as version codes and activates them if the version is requested by name. They stay active until a control line with the same name is encountered.

(e)

This signifies the end of the master version, and provokes an immediate return to the calling program.

In all cases the special character must be in columns 1 and 2, but thereafter spaces on the line are ignored. Any other line starting on stream LPC (see section 2.3) since it is probably an error.

Control lines and alternative lines whose code is not active are not copied across to the output, except as part of a set of lines between

The following is an example of a master version, holding three alternatives, and illustrating the use of version lines to alter the association of version codes with version names.

```

%%BEGIN
%%VERSION STANDARD DOUBLE=D
%%VERSION STANDARD SINGLE=S
%%VERSION IBM DOUBLE=D
S    REAL A,B
D    DOUBLE PRECISION A,B
%%VERSION IBM DOUBLE=I
    DO 20 I=1,1000
I    READ(5,10,END=30)A
S    READ(5,10)A
D    READ(5,10)A
%%VERSION IBM DOUBLE=D
    10 FORMAT(F16.6)
S    B=ALOG(A)
D    B=DLOG(A)
    WRITE(6,15)B
S  15 FORMAT(1PE12.4)
D  15 FORMAT(1PD20.12)
    20 CONTINUE
    30 STOP
    END
%%END

```

Note that it is important to establish standard names for versions of code that are ever likely to be used together.

The user must specify the required version by name, possibly with alternative names. For example he may specify IBM DOUBLE, STANDARD DOUBLE in which case he will get the Standard version if there is no IBM version. Once a name has been found, the subroutine continues to look for this name and names earlier in the list but ceases to look for names later in the list unless a effect.

The association between version name and version code is normally fixed throughout the text, but it is sometimes

convenient to change it. Where a name has occurred more than once, it is the most recent occurrence that is used. Similarly it is usual for there to be a one-one relation between names and codes, but it is also possible for several names to be associated with a single code (e.g. if all double precision versions are identical, then the code 'D' might be used for them all).

## 2.2 Argument lists and calling sequences

```
CALL OE08A( IN, OUT, NAME, LNAME, IFLAG)
```

**IN** is an INTEGER variable which must be set by the user to the stream number of the file holding the master version. It is not altered. **Restriction:**  $IN > 0$ .

**IOUT** is an INTEGER variable which must be set by the user to the stream number for output of the desired version (the device must not be a printer). It is not altered by the subroutine. **Restriction:**  $IOUT > 0$ .

**NAME** is a CHARACTER array of length **LNAME** containing the name of the desired version and alternative names in A1 format. Names are separated by commas and spaces are ignored. It is not altered.

**LNAME** is an INTEGER variable which must be set by the user to the length of array **NAME**. It is not altered. **Restriction:**  $LNAME > 0$ .

**IFLAG** is an INTEGER variable that the user need not set. After a successful entry **IFLAG** is set to zero. If any errors are found then **IFLAG** is set to an error code (see section 2.3) for the last error found.

## 2.3 Common blocks used

The subroutine contains the following COMMON block. Default values for the parameters are set by BLOCK DATA.

```
COMMON/OE08B/LP, LPC, LREC
```

**LP** is an INTEGER variable which can be set by the user to the stream number for diagnostic printing of error messages. If its value is not positive then printing is suppressed. The default value is 6.

**LPC** is an INTEGER variable which can be set by the user to the stream number for printing of faulty control lines. If its value is not positive then printing is suppressed. The default value is 6.

**LREC** is an INTEGER variable which can be set by the user to the limit on the length of the control lines to be scanned. Values of  $LREC \leq 0$  are illegal. Its default value is 72.

## 2.4 Error diagnostic values

The following errors are diagnosed:-

- 1 Value of  $IN \leq 0$ .
- 2 Value of  $IOUT \leq 0$ .
- 3 Value of  $LNAME \leq 0$ .
- 4 Version code on a
- 5
- 6 **LREC** (see section 2.3) not in the range  $12 \leq LREC \leq 72$

In all cases processing continues as far as possible, in order to find as many errors as possible.

## 3 GENERAL INFORMATION

**Use of common:** The subroutine uses common block OE08B (see section 2.3).

**Workspace:** None

**Other routines called directly:** Subroutines OE08C and OE08D are called.

**Input/output:** Input on stream **IN** and output on stream **OUT**. Error messages on unit **LP** and faulty control lines on unit **LPC**. **LP** and **LPC** have default value 6, and printing of these messages is suppressed if **LP** or **LPC** is set to 0.

**Restrictions:**

IN > 0  
IOUT > 0  
LNAME > 0

**5 EXAMPLE OF USE**

We illustrate the use of the subroutine to obtain from a master-file on input unit 1 the IBM SINGLE version and put it into output unit 2.

```
CHARACTER NAME(20)
READ(5,10)NAME
10  FORMAT(20A1)
CALL OE08A(1,2,NAME,20,IFLAG)
WRITE(6,20)NAME,IFLAG
20  FORMAT(' NAME=',20A1/' IFLAG=',I3)
STOP
END
```

The data input input is.

IBM SINGLE