

## 1 SUMMARY

This subroutine **constructs a master file, suitable for input to OE08A**, from its versions or by adding, deleting and changing versions in an existing master file.

**ATTRIBUTES** — **Version:** 1.0.0. **Types:** OE09A. **Calls:** OE15. **Original date:** December 1982. **Origin:** S. Marlow, Harwell.

## 2 HOW TO USE THE PACKAGE

### 2.1 Argument list and calling sequence

```
CALL OE09A(NFILE,NFU,NWORK,NEWMAS,IFLAG)
```

**NFILE** is an **INTEGER** variable which must be set by the user to the number of input files. This argument is not altered by the subroutine. **Restriction:** **NFILE** > 1.

**NFU** is an **INTEGER** array of length **NFILE** which the user must set to contain the Fortran unit numbers on which the files reside. The first of these files can be a master file of the form outlined in OE08A. The files are sequential with records of fixed length of which only the first 72 characters are read. This argument is not altered by the subroutine. **Restrictions:** **NFU(I) ≠ NFU(J)** and **NFU(I) > 0** for **I≠J** and **I, J=1, NFILE**.

**NWORK** is an **INTEGER** variable which must be set by the user to the Fortran unit number of a sequential workspace file, used to hold formatted records of length 80 characters. This argument is not altered by the subroutine. **Restrictions:** **NWORK > 0, NWORK ≠ NFU(I), I=1, NFILE**.

**NEWMAS** is an **INTEGER** variable which must be set by the user to the Fortran unit number of the new master file. This file must be sequential with formatted records of fixed length at least 72 characters. This argument is not altered by the subroutine. **Restrictions:** **NEWMAS > 0, NEWMAS ≠ NWORK, NEWMAS ≠ NFU(I), I=1, NFILE**.

**IFLAG** is an **INTEGER** variable which is set by the subroutine and is an error flag. On exit from OE09A, a value of zero indicates that the subroutine has performed successfully. For non-zero values see §2.4.

### 2.2 Form of the files

The first file (master file) is of the form outlined in OE08A and consists of control lines, alternative lines and ordinary Fortran lines. The other files begin with a control line of the form

```
%%VERSION name=l1l2...
```

with the remaining lines being ordinary Fortran lines. The last line of such a file can optionally be the control line with

```
%%END
```

in columns 1 to 5. Note that such a file is a trivial example of an E08A master file. The user supplies an appropriate name together with a set of letters,  $l_1l_2\dots$ , that he would like to be associated with the name in the new master file (see example in §5).

To replace a version in the master file all that is necessary is that one of the other files has the same version name. To delete a version from the master file simply have another file consisting of a version control line and an optional %%END.

### 2.3 Common Blocks Used

There are four common blocks used by OE09A. The first one OE09L allows the user to input control parameters. The other three, OE09M, OE09N and OE09O, are used for inter-subroutine communication.

The first common block can be declared as

```
COMMON/OE09L/ LP,LPC,LREC,SWITCH
```

LP is an INTEGER variable set to the Fortran unit number for diagnostic printing. Nonpositive values suppress printing. Its default value is 6.

LPC is an INTEGER variable set to the Fortran unit number for printing control lines which have incorrect syntax (an error flag being raised). Nonpositive values suppress printing. Its default value is 6.

LREC is an INTEGER variable set to the limit on the length of control lines to be scanned. It must be in the range 12 to 72. Its default value is 72.

SWITCH is an INTEGER variable with default value zero which can be ignored by the user.

The second common block has the form

```
COMMON/OE09M/ TAG(500)
```

TAG is an INTEGER array used as workspace. Its length must be at least that given by the parameter NRMAX in the labelled common area OE15G in the subroutine OE15A.

The third and fourth common blocks can be declared as

```
COMMON/OE09N/ IDUMMY(90)
COMMON/OE09O/ JDUMMY(1655)
```

IDUMMY is an INTEGER array used by the subroutine as workspace.

JDUMMY is an INTEGER array used by the subroutine as workspace and holds character information.

### 2.4 Error Diagnostics

A successful return from OE09A is indicated by a value of IFLAG equal to zero. Possible non-zero values for IFLAG are given below. In each case an identifying message is output on print stream LP (in common /OE09L/). In no case is a valid output file produced, but processing will continue, when appropriate, to look for further errors.

- 1 A version code in the master file is either not alphabetic or is the letter C.
- 2 A copy begin or copy end control line is missing.
- 3 The value of LREC is not in the range 12 to 72. In this case the value of LREC is set to 12 for a value less than 12, and 72 otherwise.
- 4 There are too many different versions.
- 5 Nonpositive Fortran unit number or there is a conflict between unit numbers.
- 6 The leading card of an ordinary file is not a valid version control card.
- 7 More than 25 version codes (letters) are necessary.
- 8 The final file needs a larger value for LREC.
- 9 There is no version control line in the first file.

### 2.5 Direct Use of OE15A

If the user increases the parameter NRMAX in the common area OE15G from its default value 500 then as well as increasing the declared sizes of arrays in the common areas of OE15 he must also increase the declared size of TAG in the common area OE09M.

### 3 GENERAL INFORMATION

**Use of common:** The subroutine uses common blocks OE09L, OE09M, OE09N and OE09O (see §2.3).

**Workspace:** None.

**Other routines called directly:** Subroutines called by the principal subroutines are OE15A (see §2.5) and the private subroutines OE09B, OE09C, OE09D, OE09E, OE09F, OE09G, OE09H, OE09I, OE09J and OE09K.

**Input/output:** Error warning and diagnostic messages on units LP and LPC.

### 4 METHOD

The first file is unpacked by OE09B into a file of records (ordinary lines) each with a 25-bit tag (stored as an integer). Each time a version name appears for the first time in a version control line, the version name is added sequentially to a table together with its version codes. If a record belongs to the *i*th version (*i*th position in the table) then the *i*th bit of the tag (considered from the least significant end) is 1, otherwise its value is zero.

Comparison is then made between this tagged file and the second file using OE15A to produce a new tagged file. The version control line is used to update the version name table and associated version codes. This comparison process is then repeated with the remaining files.

Tags are also associated with version codes (letters). If letter *l* is a version code for the *i*th version name then the tag corresponding to *l* will have a 1 in its *i*th bit. Any tagged values that occur in the final tagged file but do not have a letter are given the letters which remain. A final pass constructs an OE08A master file by using these letters in place of tags.

### 5 EXAMPLE OF USE

Suppose a user has four ordinary files containing respectively IBM SP, IBM DP, STANDARD SP, STANDARD DP versions. At the beginning of each file put respectively

```
%%VERSION IBM SP=IS
%%VERSION IBM DP=ID
%%VERSION STANDARD SP=JS
%%VERSION STANDARD DP=JD.
```

Note that we choose letters to show the similarities between versions, viz I for IBM versions, J for standard versions, S for single precision and D for double precision. If these four files were on units 10 to 13, 14 was a work file and 15 the destination file then the calling program could be

```
INTEGER NFU(4)
DATA NFU(1),NFU(2),NFU(3),NFU(4)/10,11,12,13/
NEWMAS=15
NWORK=14
NFILE=4
CALL OE09A(NFILE,NFU,NWORK,NEWMAS,IFLAG)
IF(IFLAG.NE.0)WRITE(6,10)IFLAG
10 FORMAT(14H ERROR FLAG = ,I2)
STOP
END
```