

## ELLIOTT 900 SERIES SIMULATOR

### MTALGOL PROCEDURES.

The MTALGOL procedures enable the programmer using ALGOL to write and read files and blocks of data to and from magnetic tapes. It is distributed as an RLB tape.

The procedures must be declared as code procedures within an ALGOL program that wishes to use them.

Note: The version of MTALGOL on the tape MTALGOL(ISS3).RLB is for Issue 5 ALGOL or earlier, and therefore should only be used with the load-and-go 903 ALGOL system.

Formats on tape.

All tapes start with a header block and the first two words of every block are used to hold a block number and block size (in words) respectively. The same format is used by the standard utilities, and by the SIR and FORTRAN library routines, to enable interchange of data between these systems.

### File protection and initialization.

Before a tape can be used it must be opened by the procedure mtopen. Unless the tape is named "SCRATCH" in the header block, the header block name must correspond exactly to the name used by mtopen. A tape may be prepared initially using the MTINIT utility (see 903 Utilities).

### Closing files.

Before finishing a program, all files should be closed by the procedure mtclose. If a file has been written or updated, it must be closed for writing so that an end-of-file (EOF) block is written.

## ELLIOTT 900 SERIES SIMULATOR

### Transfer of data.

Data is read and written in blocks, corresponding to the blocks physically recorded by the hardware. These are written to and from buffers in store, specified by the user as integer arrays within the program. Data is written and read by means of procedures `mtwrite` and `mtread` respectively. Real numbers may be transferred to and from buffers using the procedures `raput` and `raget` respectively.

```
code procedure  mtcheck (h, b, stat, l);  
value h; integer h, b, stat; label l; algol;
```

This procedure is used to check the operation of the other procedures and should be called after each of the other procedures before any further calls. Its exact effect depends on the individual procedure, subject to the following general rules:

1. The handler number must correspond to that used in the previous procedure, except when checking the completion of a rewind in `mtclose`.
2. If the previous procedure was successful, the exit from `mtclose` is normal, parameter `b` will hold the block number of the last block read or written and `stat` will hold the status word read from the magnetic tape controller on completion of the operation. If the previous procedure was not successful, `mtcheck` exits to the label `l` and `b` contains an error number. `stat` again holds the status word, which will indicate the most recent hardware error state.

Error codes are:

1. Repeated hardware error (in the simulator can only arise if a tape file has not previously been closed).
2. File not opened as names incompatible.
3. Specified block cannot be found.
4. Handler in manual or otherwise not available.
5. Instruction rejected as "do nothing" (due to hardware or manual interference - cannot arise in the simulator using the standard libraries).
6. Attempt to read or write beyond the physical end of tape.
7. File not opened for reading or writing before an attempt to read or write respectively.
8. Write permit ring not fitted when attempting to write.
9. Long block - the block read from tape is longer than the buffer allocated.

## ELLIOTT 900 SERIES SIMULATOR

Other values of b can occur which are not error, but still branch to the label 1 as special action may be required:

- 0 The end of tape marker has been detected, the file should be closed for writing
- 1 The block specified to be read is a label block, other than an EOF block.
- 2 The block specified to be read is an EOF block.
- 3 The handler is still busy rewinding after a call of mtclose.

```
code procedure mtopen (h, a, t, s);  
value h, t; integer h, t; string s; integer array a; algol;
```

This procedure is used to open a file on a magnetic tape, mounted on handler h (h=0,1,2,3). s gives the name of the file to open. Only the first 12 characters, not including the opening and closing quote symbols, are significant. a is an integer array used as a buffer and must be at least 56 elements long. T indicates the type of operation required:

- 1 open file for reading only. The name of the file must correspond exactly to the name given by s. The tape is positioned ready to read the first data block.
- 0 open file for updating. The name of the file must correspond exactly to the name given by s. The tape is positioned ready to read or write the first data block.
- 1 open file for writing. The original name on the file must correspond to s, or be 'SCRATCH'. A[28] to a[55] must be set up as described below.

If the file is opened for updating the existing header is checked but not overwritten. If information is to be added to the end of the file, the program should read forward until the EOF block is detected, go back one block and start writing.

If the file is opened for writing, a new header is written and a complete new file must be written.

In each case after a call of mtcheck, the original header will be left in a[0] to a[27] (assuming the lower bound of a is 0). When open for writing, the header block is written from locations a[28] to a[55]. These locations should be set to zero, or appropriate values for header block locations 0; to 27; before mtopen is called. (See SIR magnetic tape routines for details of header format). Optionally the name of the file may be specified as packed SIR internal code in elements a[30] to a[34], in which case s should be the empty string ''.

## ELLIOTT 900 SERIES SIMULATOR

```
code procedure mtwrite (h, a, p);  
value h, p; integer h, p; integer array a; algol;
```

This procedure is used to write a block of data, or a label block to tape. h is the handler number required. a must hold the block to be written: the first two words must not contain significant data as they will be overwritten when the block is written to tape. P should be set to the number of words to be written, including the two locations at the start of the buffer. The call of mtcheck that should follow mtwrite will give the number of the block just written. If blocks are to be written other than following the last block accessed, mthead should be used to position the tape. If a block is written in this way, further blocks along the tape will become unreadable: it is not possible to update blocks at random. If a label block rather than a data block is to be written, p should be given the value (number of words-131072).

```
code procedure mthead (h, a, n);  
value h, a; integer h, b; integer array a; algol;
```

This procedure is used to find and read a block on a tape. N is the required block number. If n is 0 the next block in sequence is read. Since all blocks are numbered sequentially the selected block can be found by scanning up and down a tape. h is the handler number required. a is an integer array used as a buffer. After reading and checking with mtcheck the block specified by n will be held in a. The first two words will contain the standard block header, in particular the second word will contain the block length. The search will start with a rewind or number of backspace movements as appropriate, then will continue forward until the block is found. Note if the block read is a label block, mtcheck will exit via the label l, but the label block content will be in a and the block length will be in the second location of a in the form (length-131072).

```
code procedure mtclose (h, t);
```

This procedure is used to close a magnetic tape file: files that have been written or updated must be closed after the final block has been written so that a correct EOF block can be added to the end of the file. h is the handler number required. t specifies the type of operation required:

0. Rewind tape without closing file.
1. Close file for writing. Write an EOF label and rewind.
2. Close file for reading and writing. Write an EOF label and rewind.

## ELLIOTT 900 SERIES SIMULATOR

3. Close file for reading only. Rewind tape (only use if file not opened for writing).

After closing for writing ( $t = 1$  or  $2$ ) `mtcheck` must be called before any other references to any of the tapes. After this (and any other all of `mtclose`) `mtcheck` may be used at any time to check whether the rewind operation is complete.

```
code procedure raput (a, i, r);  
value i, r; integer array a; integer i; algol;
```

Store the value `r`, in packed form, in the  $(i+1)$ th and  $(i+2)$ th locations of `a`; ALGOL error number 5 is reported if `i` is out of bounds.

```
code real procedure raget (a, i);  
value a, i; integer array a; integer i; algol;
```

take the real value packed into the  $(i+1)$ th and  $(i+2)$ th locations of `a`, as stored by `raput`. There is no check on the value of `i`.

Note: The Elliott programming documentation gives the impression that `raput` and `raget` are part of the MTALGOL library, but they are not included in any of the MTAGOL paper tapes found by the author. It may be they were included in a later issue only distributed on magnetic tape and now lost. Code procedures corresponding to the above specification have been supplied by the author as a separate library tape (RAGETPUT.RLB in directory 903ALGOL).