M20

OLISORT
User Guide

olivetti L1

M20

OLISORT
User Guide

olivetti L1

PREFACE

This manual describes the Olisort Sort/Merge package for new users. As far as possible, the manual is self contained. If you are not a BASIC programmer you can still use Olisort with the aid of utilities provided with the package. We recommend that new users read Chapters 1 to 4 before doing anything else.

SUMMARY

After an introduction we take a look at the concepts and facilities of Olisort. In Chapter 3 we explain the procedures, related to the M20, that you will need to know to use Olisort. Chapter 4 contains a detailed description of the parameters which you pass to Olisort in order to drive it. In Chapter 5 we look at the way you can call Olisort from a BASIC program, and in Chapter 6 we talk about the interactive utility, olisortx, which allows you to use Olisort without needing any knowledge of BASIC programming. The final chapter is a discussion of the utility olisortp, with which you can build parameter files.

A Reference Card summarising all the information needed to use Olisort is provided for experienced users.

REFERENCES:

Professional Computer Operating System (PCOS) - User Guide
Code 3930090 C

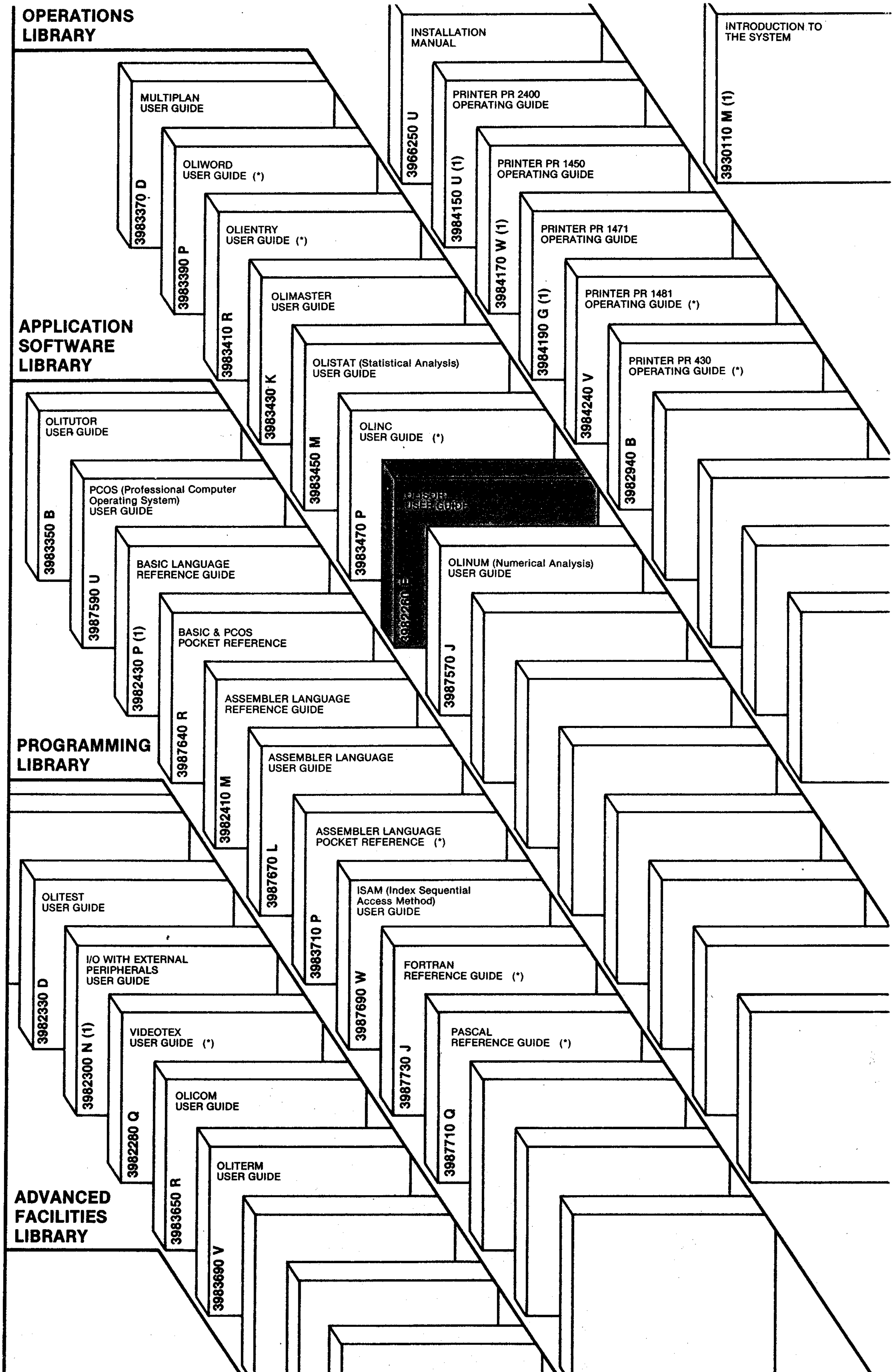
BASIC Language - Reference Guide
Code 3982430 P(1)

DISTRIBUTION: General (G)

FIRST EDITION: October 1982

PUBLICATION ISSUED BY:

Ing. C. Olivetti & C. S.p.A.
Servizio Centrale Documentazione
Via Jervis, 77 - IVREA (Italy)



OPERATIONS LIBRARY

INSTALLATION MANUAL 3966250 U

PRINTER PR 2400 OPERATING GUIDE 3984150 U (1)

PRINTER PR 1450 OPERATING GUIDE 3984170 W (1)

PRINTER PR 1471 OPERATING GUIDE 3984190 G (1)

PRINTER PR 1481 OPERATING GUIDE (*) 3984240 V

PRINTER PR 430 OPERATING GUIDE (*) 3982940 B

INTRODUCTION TO THE SYSTEM 3930110 M (1)

APPLICATION SOFTWARE LIBRARY

MULTIPLAN USER GUIDE 3983370 D

OLIWORD USER GUIDE (*) 3983390 P

OLIENTRY USER GUIDE (*) 3983410 R

OLIMASTER USER GUIDE 3983430 K

OLISTAT (Statistical Analysis) USER GUIDE 3983450 M

OLINC USER GUIDE (*) 3983470 P

OLISORT USER GUIDE 3982260 E

OLINUM (Numerical Analysis) USER GUIDE 3987570 J

PROGRAMMING LIBRARY

OLITUTOR USER GUIDE 3983350 B

PCOS (Professional Computer Operating System) USER GUIDE 3987590 U

BASIC LANGUAGE REFERENCE GUIDE 3982430 P (1)

BASIC & PCOS POCKET REFERENCE 3987640 R

ASSEMBLER LANGUAGE REFERENCE GUIDE 3982410 M

ASSEMBLER LANGUAGE USER GUIDE 3987670 L

ASSEMBLER LANGUAGE POCKET REFERENCE (*) 3983710 P

ISAM (Index Sequential Access Method) USER GUIDE 3987690 W

FORTRAN REFERENCE GUIDE (*) 3987730 J

PASCAL REFERENCE GUIDE (*) 3987710 Q

ADVANCED FACILITIES LIBRARY

OLITEST USER GUIDE 3982330 D

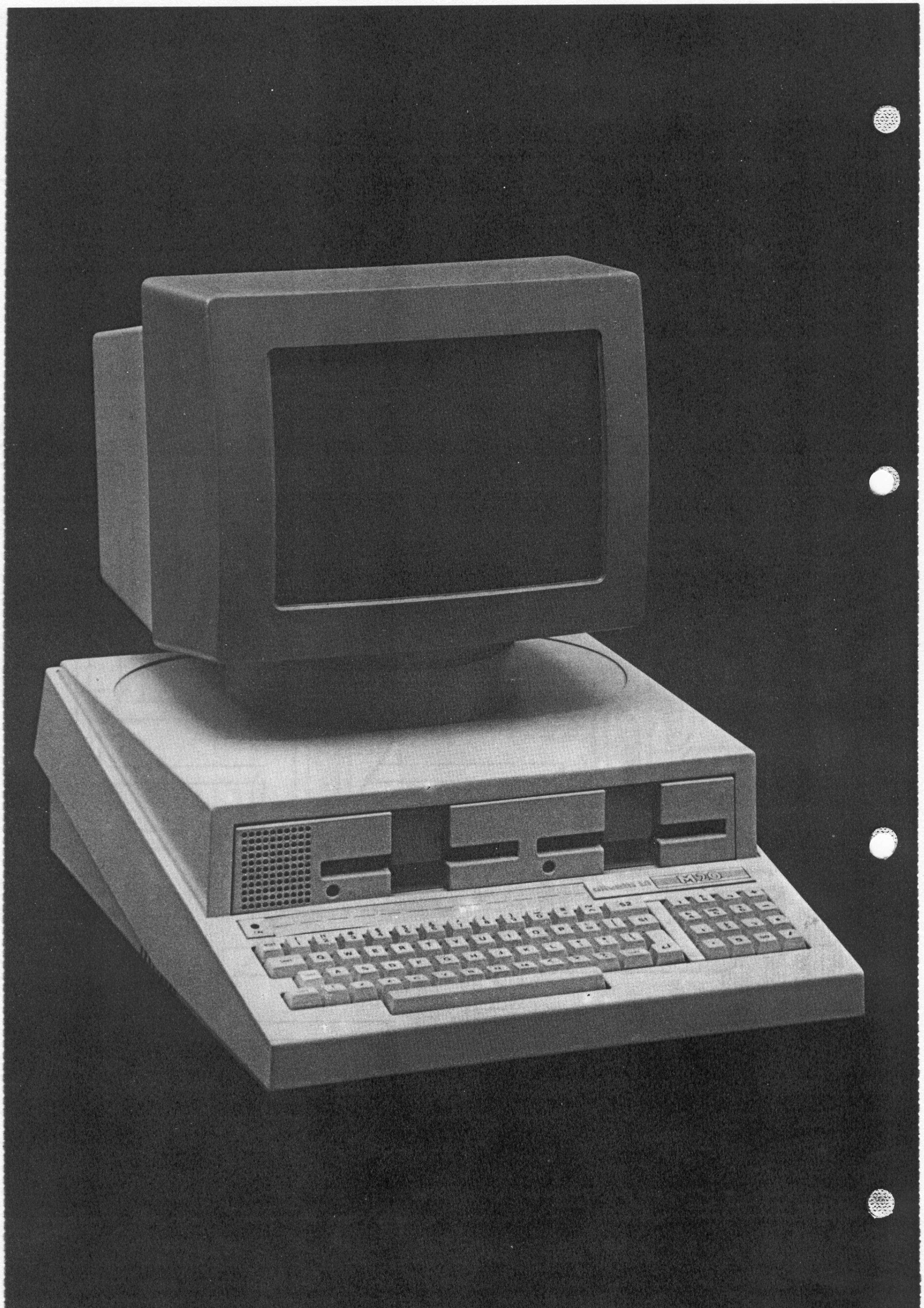
I/O WITH EXTERNAL PERIPHERALS USER GUIDE 3982300 N (1)

VIDEOTEX USER GUIDE (*) 3982280 Q

OLICOM USER GUIDE 3983650 R

OLITERM USER GUIDE 3983690 V

(*) This publication will be made available with the next release.



M20

CONTENTS

1. INTRODUCTION

ABOUT THIS CHAPTER

Here we give a brief outline of Olisort and describe the physical contents of the Olisort Package. We also tell you about the PCOS system disk, which you may need to use, and about the notation we have adopted throughout this manual.

CONTENTS

INTRODUCTION

INTRODUCTION TO OLISORT

Olisort is a versatile, high-performance, sort/merge package, tailored to run on the Olivetti M20 microcomputer. Its wide range of facililties make it useful in almost all types of business application.

With Olisort, you can sort a file, merge two files together, select records from a file, append one file on to the end of another, or sort and select at the same time. This depends on which Olisort Mode you use.

Olisort is driven by parameters that are passed in a Parameter String. The entire Parameter String can be passed to Olisort in a Command String written into a BASIC program, or via a short Command String which makes reference to a Parameter File on disk. A Parameter File or a Command String can be created interactively, via the Olisort utilities. This means you can very easily create lengthy Parameter Strings, while appreciably reducing the possibility of keying errors.

You can run Olisort by calling it from your BASIC program, or by using the utility olisortx. This allows you to build more than one sort into a suite of programs that runs as a continuous job, or to take a particular file and sort it immediately.

The major internal features of Olisort are as follows:

- Up to 10 sort keys per record can be specified.

- Selection or Exclusion of records is allowed via, up to four, Select/Exclude keys.

- Select/Exclude keys can be combined together via the logical connectors "AND" or "OR".

- For each individual Select/Exclude key you can specify the comparative operators, "less than", "equal to", or "greater than".

- For each individual Select/Exclude key you can use Wild-Card Characters within comparisons.

- With Sort keys, and Select/Exclude keys, you can choose to take lower case letters and upper case letters as the same alphabetic value.

- Disks used for the Olisort work files and for the output files can be changed, and Olisort will supply the prompts for disk changes at the appropriate time.

- You may skip up to 32,767 records of a file before beginning to sort or merge.

- Comprehensive error handling facilities are written into Olisort, and the status code is always returned to the program or utility which calls Olisort.

- Incorporating Olisort into existing BASIC programs is very simple and requires no knowledge of assembler interfacing.

Olisort is specifically designed to run under the M20 BASIC Interpreter and supports standard fixed-length records, with fixed-length fields.

THE OLISORT PACKAGE

The Olisort package is contained on a single disk. It consists of the following files:

- os.cmd which is used to load the Olisort program into memory

- olisort.bas is the Olisort Interface Program, with which you can call Olisort from a BASIC program

- olisortx.bas which you can use to interactively create Command Strings and automatically initiate Olisort

- olisortp.bas with which you can build Parameter Files

- OLISORT.ABS which is the Olisort program image.

The original disk should not be used, but kept as a master copy. You can make yourself a working Olisort disk by following the backup procedure described in Appendix B. From now on we will refer to the working copy of your original disk as the OLISORT disk. Please note that you will find it much easier to make a backup copy after you have read Chapter 3.

It is also advisable to become familiar with the method of physically Write Protecting disks, before starting to use the Olisort package. This is explained in Appendix A.

You may find the original Olisort disk in the envelope inside the back cover of this manual. If not you can obtain it from your Olivetti dealer. In the same envelope you should find the Olisort Reference Card, which summarises the workings of Olisort.

THE SYSTEM DISK

A PCOS System Disk is always supplied with your M20. You will need to use it to load PCOS commands into memory, to make backups, for formatting, etc. The disk is illustrated in Figure 1-1.

Disk No.
Mini FD 220
olivetti
Cod. 82276T
9780447J
© PCOS Rel.
© BASIC-8000
Copyright © by Ing. C. Olivetti & C. S.p.A. · 1981
olivetti
Mini FD 220

Figure 1-1 PCOS System Disk

NOTATION CONVENTIONS USED THROUGHOUT THIS MANUAL

When we want you to key in information, we use a particular notation convention to represent the keys of the M20 keyboard. This notation convention is based on the use of white-on-black characters to represent individual keys. For example the keys A, 1, ", ?, are represented as follows:

A 1 " ?

Each key you have to press is separated by a space from the next one. This does not mean you have to press the space bar. If we want you to enter a space we print SPACE.

As you have probably noticed, all the alphabetic keytops of the M20 are marked with upper case letters. This doesn't mean that you get an upper case letter when you press the key. In fact the opposite is true. If you press any of the alphabetic keys you obtain a lower case letter.

To obtain an upper case letter you have to hold down the SHIFT key while you are pressing the alphabetic key. Within the notation used in this manual, when we want you to enter an upper case letter, we show the letter in upper case.

For example, if you should enter 'small letters', we would show:

s m a l l SPACE l e t t e r s

and if you should enter "BIG LETTERS", we would show:

B I G SPACE L E T T E R S

To produce the phrase BIG LETTERS, you would have to hold down the SHIFT key while you pressed each key, or use the shift lock, as described later in this section.

For most of the keys we represent in this way, it is obvious which key on the M20 keyboard we mean you to press. However there are those that are not so obvious. The list below should clarify all cases where there might be some doubt.

COMMAND means the yellow COMMAND key.

CTRL means the blue CTRL key.

SHIFT means one of the two SHIFT keys.

SPACE means the space bar.

CR means the L-shaped end-of-line key, with the arrow on it.

When it is important to differentiate between the letter O and zero, we put a slash through the zero. Like this: Ø. We also do this when we want you to key in zero. Like this: Ø.

If we want you to press two keys at the same time we put a plus sign between them. For example if we wanted you to press the CTRL key and the C key, we would show:

CTRL + C.

ERROR MESSAGES

While using the procedures described in this manual you may make errors. If so, you may find an error message on the screen. A full list of these messages is given in Appendix D.

Often you will find that you have made a simple keying error which can be corrected by keying in the information again.

2. OLISORT CONCEPTS AND FACILITIES

ABOUT THIS CHAPTER

Here we look at the concepts which were used in creating the Olisort Package, and the facilities which it makes available to you. This chapter is a tutorial overview of Olisort, and parts of it may be useful as a reference while you are becoming familiar with the use of Olisort.

CONTENTS

OLISORT CONCEPTS AND FACILITIES

RUNNING OLISORT

There are two methods of running Olisort:

- By incorporating the parameters into a Command String in a BASIC program, as described in Chapter 5.

- By using the olisortx utility to interactively create parameters and then run Olisort, as described in Chapter 6.

In either case the parameters passed have the same format.

PASSING PARAMETERS TO OLISORT

There are two ways of passing parameters to Olisort:

- Through a Parameter File saved on disk. This is very useful when you want to run the same sort more than once. You can create a Parameter File by using the olisortp utility as described in Chapter 4, under the Mode 3 heading.

- Through a Command String which is a BASIC variable with the name OLISORT.COM$ you can create a Command String in two ways:

  . By using the BASIC interpreter to create a Command String as described in Chapter 5.

  . By using olisortx to create a Command String which you can store on disk, as described in Chapter 6.

It should be noted that, even if you use a Parameter File for passing parameters, you still have a short Command String in your BASIC program. This Command String contains only a few parameters and a reference to the Parameter File that contains the majority of the parameters.

OLISORT MODES

One of the parameters you have to pass to Olisort specifies the Mode you want to use for the sort. There are four Olisort Modes.

OLISORT MODE Ø

Mode Ø allows you to Sort a file, Select and Exclude records from a file, or Sort and Select/Exclude at the same time. The parameters for the Sort, Select, or Sort and Select/Exclude, must already exist in a Parameter File on disk.

For example if we have a master file containing stock items in part number order, and two transaction files containing new parts from two different sources, we could sort the transaction files in the following way:

- Create a Parameter File via olisortp, in order to sort the transaction files.

- Write a BASIC program to call Olisort twice using the same Parameter File each time, to sort the two transaction files. This involves making a Command String and specifying Mode Ø. This Command String would be very short, as most of the parameters would be contained in the Parameter File on disk.

This is illustrated in Figure 2-1.

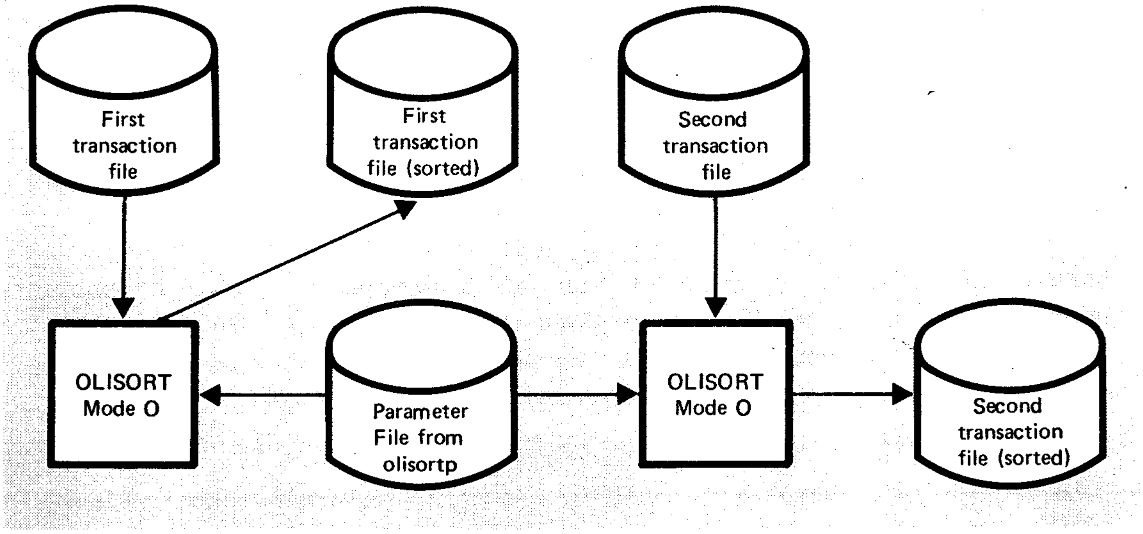

First transaction file
First transaction file (sorted)
Second transaction file
OLISORT Mode O
Parameter File from olisortp
OLISORT Mode O
Second transaction file (sorted)

Figure 2-1 Sorting in Mode Ø

OLISORT MODE 1

Mode 1 will merge two sorted files into one, or append one file onto the end of another. Once again a Parameter File must have been created on disk. If the two files have been sorted and the Sort Key information is specified in the parameter file, a merge will be performed. If no Sort Key information is specified in the Parameter File, the two files will be appended.

For example, if we think about the transactions files and master file we talked about for Mode Ø, we could perform the master file update as follows:

- Using the same Parameter File that we created to sort the two transaction files, create a BASIC program containing a Command String which specifies Mode 1. We specify the two input files as our two sorted transaction files.

- When the BASIC program has been run the two transaction files will be merged into one transaction file.

- Still using the same Parameter File, create a BASIC program containing a Command String which specifies Mode 1. This time our input files are merged transactions and the master file.

- When the BASIC program has been run an updated master file is produced.

This is illustrated in Figure 2-2.

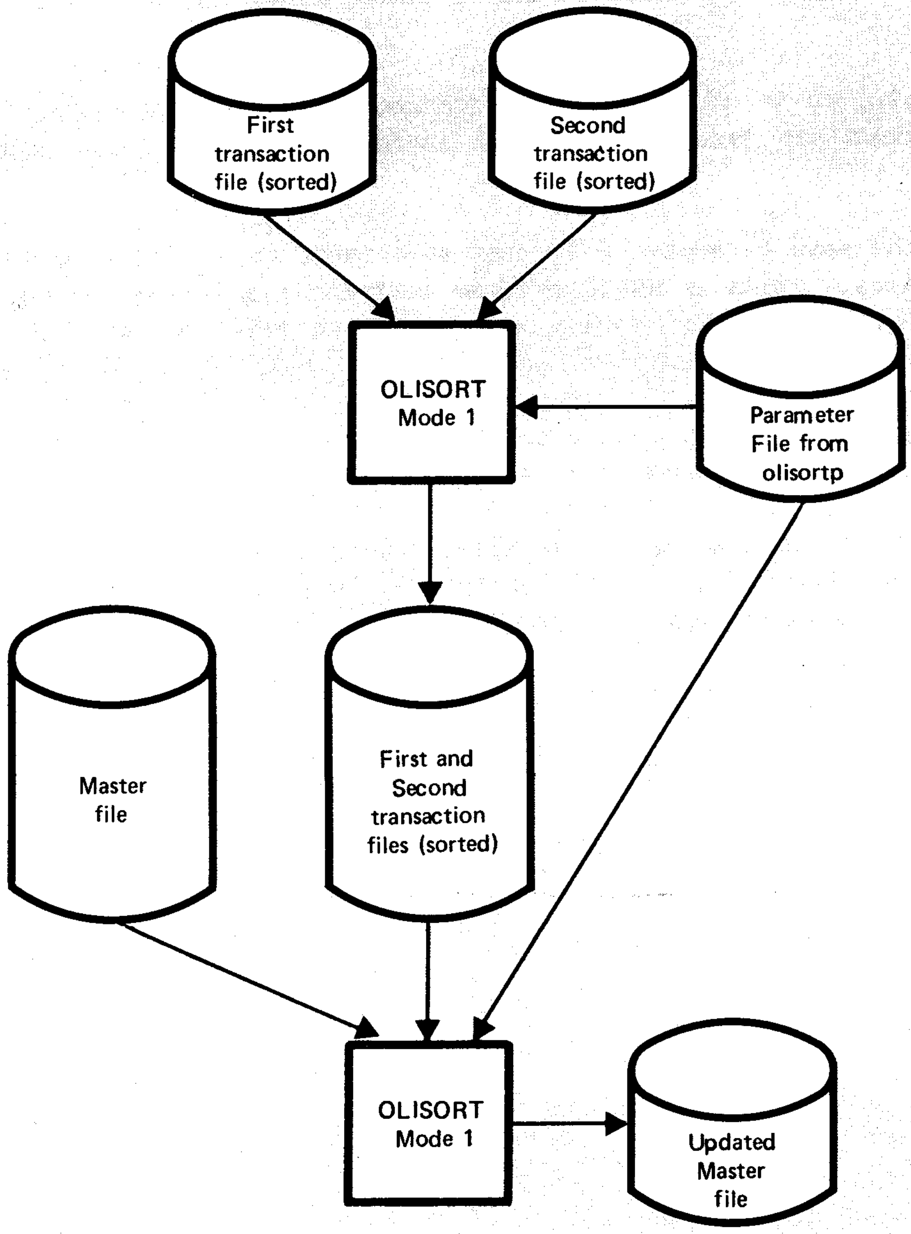

First transaction file (sorted)
Second transaction file (sorted)
OLISORT Mode 1
Parameter File from olisortp
Master file
First and Second transaction files (sorted)
OLISORT Mode 1
Updated Master file

Figure 2-2 Merging in Mode 1

OLISORT MODE 2

Mode 2 performs the same functions as Mode Ø, but all the parameters are passed in a Command String written into a BASIC program.

For example, we could sort one of the transaction files we talked about in Mode Ø, in the following way:

- Write a BASIC program containing a Command String which specifies Mode 2.

- When the BASIC program is run, a sorted transaction file will be produced.

This is illustrated in Figure 2-3.

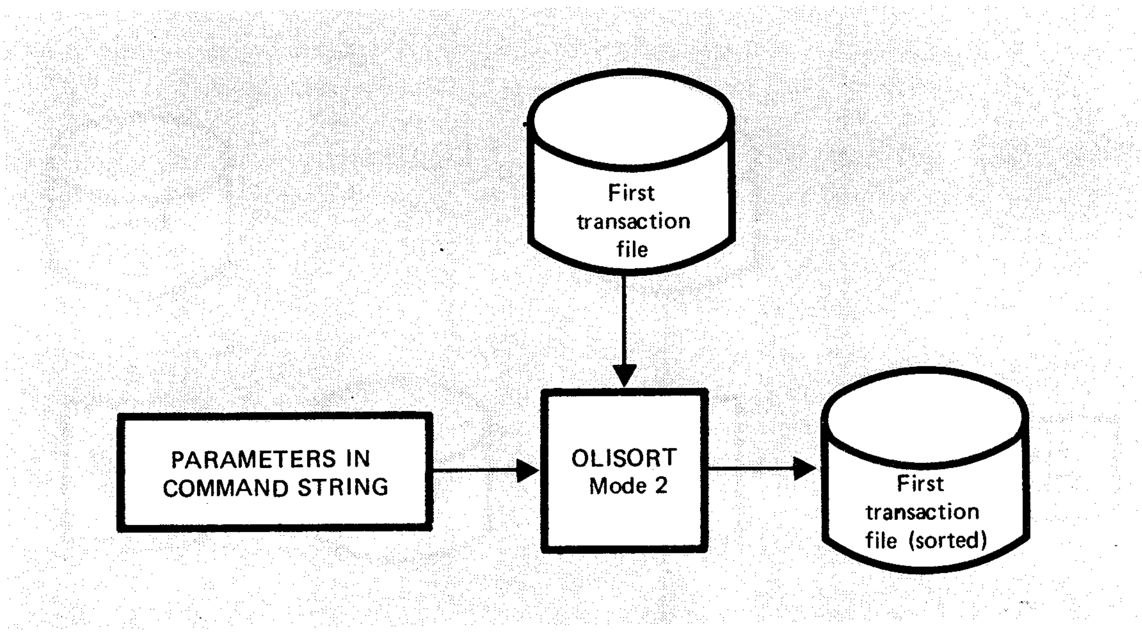


Figure 2-3 Sorting in Mode 2

OLISORT MODE 3

Mode 3 is identical to Mode 2, except that a Parameter File is created on disk from the parameters set up in the Command String. This facility is useful if you have written a Command String into a BASIC program and wish to produce a Parameter File from it.

For example, we could sort our two transaction files from Mode Ø in the following way:

- Write a BASIC program containing a Command String which specifies Mode 3. The input file for this sort is the first transaction file.

- When the BASIC program is run, the first transaction file is sorted, and a Parameter File is produced.

- We then write another BASIC program containińg a Command String which specifies Mode Ø and uses the Parameter File produced in the last step. This time the input file is the second transaction file.

- When the program is run, the second transaction file is sorted.

This is illustrated in Figure 2-4.

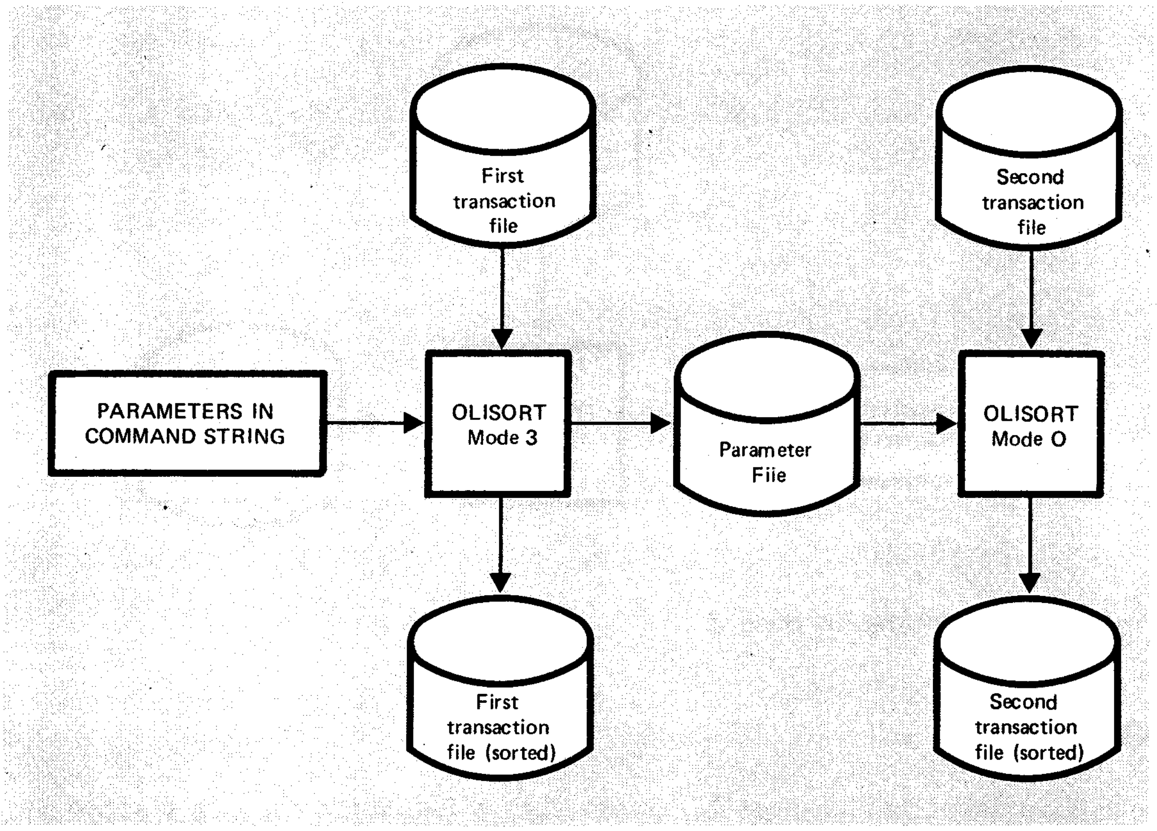

First transaction file
Second transaction file
PARAMETERS IN COMMAND STRING
OLISORT Mode 3
Parameter File
OLISORT Mode O
First transaction file (sorted)
Second transaction file (sorted)

Figure 2-4 Sorting in Mode 3

FILE AND RECORD SIZE

The maximum file size which can be input to Olisort is limited by thĕ available disk space since multiple-volume files cannot be used. The limiting factor is the size of the Sort Work-File which can be up to 1.5 times as great as the Input File(s) if 1Ø Sort Keys and 4 Select/Exclude Keys are specified.

The maximum record length, allowed by default, is 256 bytes. In the case of files with exceptionally long records this figure can be altered. In order to do this you have to use the PCOS command SBASIC, which is described in the 'Professional Computer Operating System (PCOS) User Guide', code. For simplicity throughout this document we refer to the maximum record length as being 256 bytes.

OLISORT FIELD TYPES

All files must contain fixed-length records, with fixed-length fields.

Within Sort Keys and Select/Exclude Keys you can specify the following types of field:

- Alphanumeric Text Strings

- Hexadecimal Fields

- Integer Numbers

- Single Precision Numbers

- Double Precision Numbers

If you are specifying an Alphanumeric Text String in a Sort Key or Select/Exclude Key, you are telling Olisort to take the same upper case and lower case letters as the same value.

If you specify Hexadecimal Fields in a Sort Key or Select/Exclude Key, you are telling Olisort to treat characters as hexadecimal codes. This means the same upper and lower case letters will have different values.

SORT FEATURES

When sorting you can specify up to 1Ø Sort Keys and have the sort performed in ascending or descending order on each of the individual keys.

For instance you could sort a customer file in descending order of outstanding balance, within ascending order of state, to produce a listing like the one illustrated in Figure 2-5.

STATE	CUSTOMER	O/S BALANCE ($)
Nevada	M & M Ltd.	2596.00
	Zilch & Sons	1976.55
	ABC Co.	568.88
New Jersey	NXR Ltd.	8870.77
	Wyatt & Hyatt	8345.98
	Acme Ltd.	5689.00
	Olive Oil Co.	67.43

Figure 2-5 Outstanding Balances List

You can sort to produce a new sorted output file, or overwrite the unsorted input file with the sorted output. This is a useful facility if disk space is at a minimum, and you do not want to change disks during the sort. However, if a machine failure occurs during the sort, you could lose your input file. If you do decide to overwrite the input file, the simplest insurance against machine failure is to make a backup copy of the disk which contains the input file. You can find out how to make a backup copy in Appendix B.

SELECT/EXCLUDE FEATURES

With Olisort you can produce an output file which contains only selected records from the input file. You can perform a Select/Exclude independently, or at the same time as you perform a Sort. For example, if we decided to perform a Select/Exclude and a Sort on the customer file we used in the last section, choosing to exclude any customers whose balance was less than $1000, we would produce a printout like the one illustrated in Figure 2-6.

STATE	CUSTOMER	O/S BALANCE ($)
Nevada	M & M Ltd.	2596.00
	Zilch & Sons	1976.55
New Jersey	NXR Ltd.	8870.77
	Wyatt & Hyatt	8345.98
	Acme Ltd.	5689.00

Figure 2-6 Outstanding Balances List with Select/Exclude

Records can be selected or excluded by specifying up to four Select/Exclude keys. These keys are completely independent of the Sort Keys which may be specified at the same time. That means you can select or exclude on different fields from those you are sorting on. If you specify only Select/Exclude keys and no Sort Keys, Olisort will produce an output file containing selected records, but in the same order as the input file.

If you specify Alphanumeric Text Strings within a Select/Exclude Key, Olisort will ensure that the contents of the Select/Exclude Key and the field within the record are the same length, before comparing them. the logic behind this procedure is as follows:

- First Olisort checks to see which is the longer, the Select/Exclude Key or the field it is to be compared with.

- The longer one is truncated, for the purposes of comparison, to the length of the shorter one.

- The comparison then takes place, character by character, and if all the characters are the same, there is a match.

For example if we have a field in a record which looks like this:

LEAD PIPE 12mm x 3m

and we want to exclude all forms of LEAD PIPE from our output file, we could use specify exclusion and use a Select/Exclude Key that looks like this:

LEAD PIPE

WILD-CARD CHARACTERS

You can put in your Select/Exclude Key, what are called Wild-Card Characters. These are as follows:

< in the Select/Exclude Key means that the Select/Exclude Key character will be considered less than the corresponding character in the field under comparison.

> in the Select/Exclude Key means that the Select/Exclude Key character will be considered greater than the corresponding character in the field under comparison.

= in the Select/Exclude Key means that the Select/Exclude Key character will be considered equal to the corresponding character in the field under comparison.

The use of Wild-Card Characters is connected with the specification of the LEG parameter for Select/Exclude Keys. Via the LEG parameter, you can specify that selection or exclusion takes place if the Select/ Exclude Key is Less than, Equal to, or Greater than, the field under comparison. By using Wild-Card Characters you can do partial or generic key comparisons.

For example, if we have a parts file which contains part numbers of the type illustrated in Figure 2-7, and we want to do a partial selection of parts belonging to Class A, which have an Assembly Code of 8Ø, we could use a Select/Exclude Key which looks like this:

8Ø======A

ASSEMBLY CODE	PART NUMBER	PART CLASS
94	543678	B
67	553478	A
8Ø	88769Ø	A
88	113425	D
8Ø	9844ØØ	C
8Ø	ØØØ789	A
56	72299Ø	B

Figure 2-7 Parts File Extract

From the part numbers shown in Figure 2-7, using our Select/Exclude Key, we would select the following:

8Ø-88769Ø-A
8Ø-ØØØ789-A

USING MULTIPLE SELECT/EXCLUDE KEYS

If you use more than one Select/Exclude Key when selecting or excluding records, you have to combine them together with the logical operators "AND" or "OR". Up to four Select/Exclude Keys can be specified, which means you can use up to three logical connectors.

If you use AND to combine two Select/Exclude Keys, both of the Select/Exclude Keys must match with the corresponding fields for the selection to take place.

If you use OR to combine two Select/Exclude Keys, only one of these Keys must match with the corresponding field for the selection to take place.

For multiple Select/Exclude Keys, where exclusions are specified, the logical connector is always taken to mean OR. In other words, if we have two Select/Exclude Keys joined by the logical connector AND, a match with the record and either one of them is enough for exclusion to take place.

For example, if we used the following Select/Exclude Key combination, specifying selection, on the part numbers illustrated in Figure 2-3:

8Ø======A OR 8Ø======C

we would select the following part numbers:

8Ø-887692-A
8Ø-9844ØØ-C
8Ø-ØØØ789-A

Whereas, if we used the following Select/Exclude Key combination on the part numbers illustrated in Figure 2-3:

8Ø======A AND 56======B

we would not select any part numbers.

When using multiple Select/Exclude Keys and specifying exclusion and selection together, the Select/Exclude Keys associated with exclusion are examined first. If a match situation does not occur, the Select/Exclude Keys associated with selection are examined. If a match situation does occur for the Select/Exclude Keys associated with exclusion, the exclusion of the record under comparison will take place.

MERGE FEATURES

Olisort allows you to merge two sorted input files into one output file, or to append one input file on to the end of the other.

In order to merge two input files together, you have to specify the same Sort Keys that you would use to sort the two input files.

To append two input files, you don't specify any Sort Keys. The second input file specified is then automatically appended on to the end of the first input file specified.

By using repetitive merges you can quickly produce a single output file from several, sorted, input files.

CHANGING DISKS

You can specify, via suitable parameters, that you want to change disks during a Sort, Merge, Select/Exclude, or Sort and Select/Exclude. Olisort automatically supplies the operator prompts for the disk changes.

You can change disks on which Olisort writes:

- its Sort Work-Files

- the output file, as long as you haven't specified that the output file is to overwrite the input file, for a Sort procedure.

For detailed information on disk changes see Appendix F.

FILES USED BY OLISORT

Olisort may use up to five files, while running. The files are as follows:

- INPUT1 which is either: the only input file for a Sort, Select/Exclude, or Sort and Select/Exclude; or the first input file for a Merge or Append.

- INPUT2 which is the file produced when the input file(s) has(have) been Sorted Select/Excluded, Sort and Select/Excluded, Merged, or Appended. For Sort only, you can specify that the output file overwrites the input file INPUT1. This saves disk space, but introduces the possibility of losing INPUT1 if there is a power or machine failure. The best insurance against losing your input file is to make a backup of the disk which contains it. The method of making backups is described in Appendix B.

- BDUMP which is a file created by Olisort, and used to dump the BASIC Interpreter during the processing. This is a means of optimizing memory within the M20 and increasing the efficiency of Olisort. When Olisort has finished processing, the BASIC Interpreter is automatically reloaded into memory and the BDUMP file is erased. The BDUMP file needs 32,768 bytes of disk space.

- SWORK is the Sort Work-File created by Olisort. It can be up to 1.5 times larger than the Input File(s). When Olisort has finished its processing, the Sort Work-File is automatically erased.

3. STARTING UP OLISORT

ABOUT THIS CHAPTER

Here we tell you how to start the M20, load disks, and get in the PCOS or BASIC environment. We also show you how to list the files on a disk.

This information will help you to use Olisort.

If you are already familiar with the M20, this chapter may be skipped.

CONTENTS

STARTING UP OLISORT

POWER ON/OFF

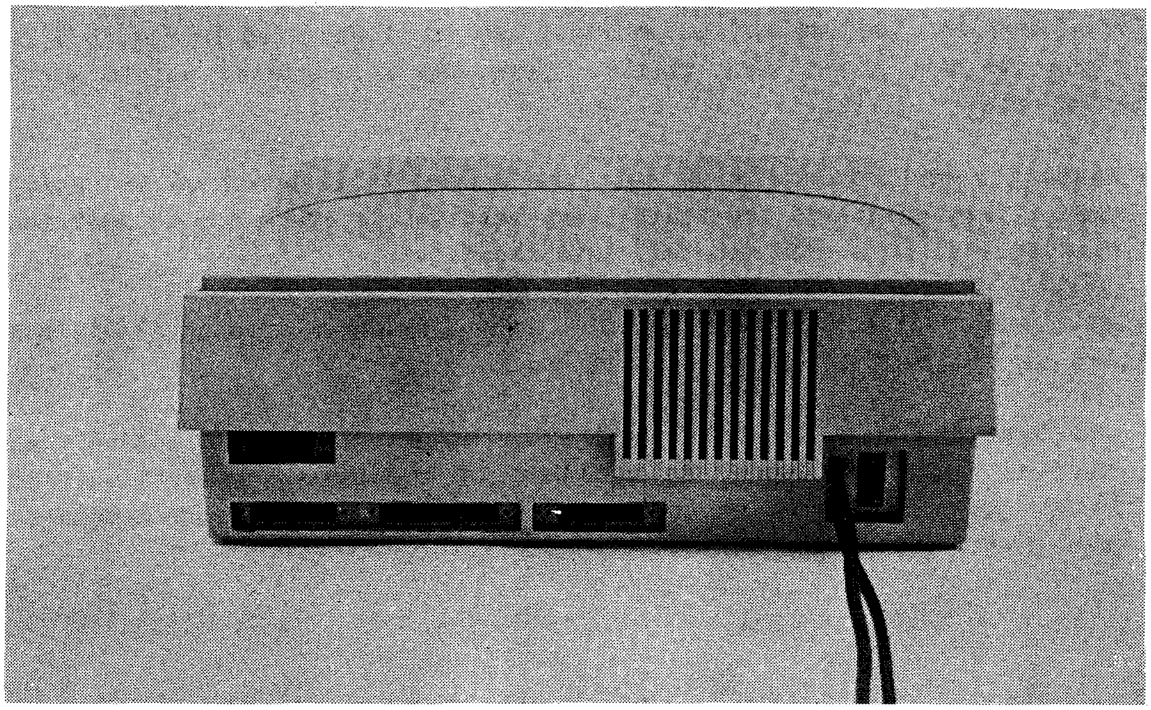


Figure 3-1 The M20 ON/OFF Switch

Before you can do anything else you have to power on the M20. This is a very simple procedure. Press the ON/OFF switch on the rear of the machine to the 1-position. After a short delay a message similar to the following one appears:

```
Bootstrap Loader Rev 1.Ø
Insert Diskette and type return
```

The M20 is now active and the required disks can be loaded, as described under the "Loading Disks" heading.

To power off the M20, simply remove any disks that are loaded and press the ON/OFF switch to the Ø position.

LOADING DISKS

The method for loading and unloading disks is illustrated in Figure 3-2. As you can see it is very simple. There are a couple of points worth noting here:

- never try and force a disk into the drive. If it won't go in easily take it right out and try again. Very soon you'll learn how to seat the disk so that it slides smoothly into the drive.

- When you do insert a disk into the drive, you have to push it all the way in, until it clicks into position.

- When removing disks from the drive you should be careful to ensure that the red indicator light, to the left of the drive, is extinguished before you open the drive cover. Failure to do so may cause damage to your disk.

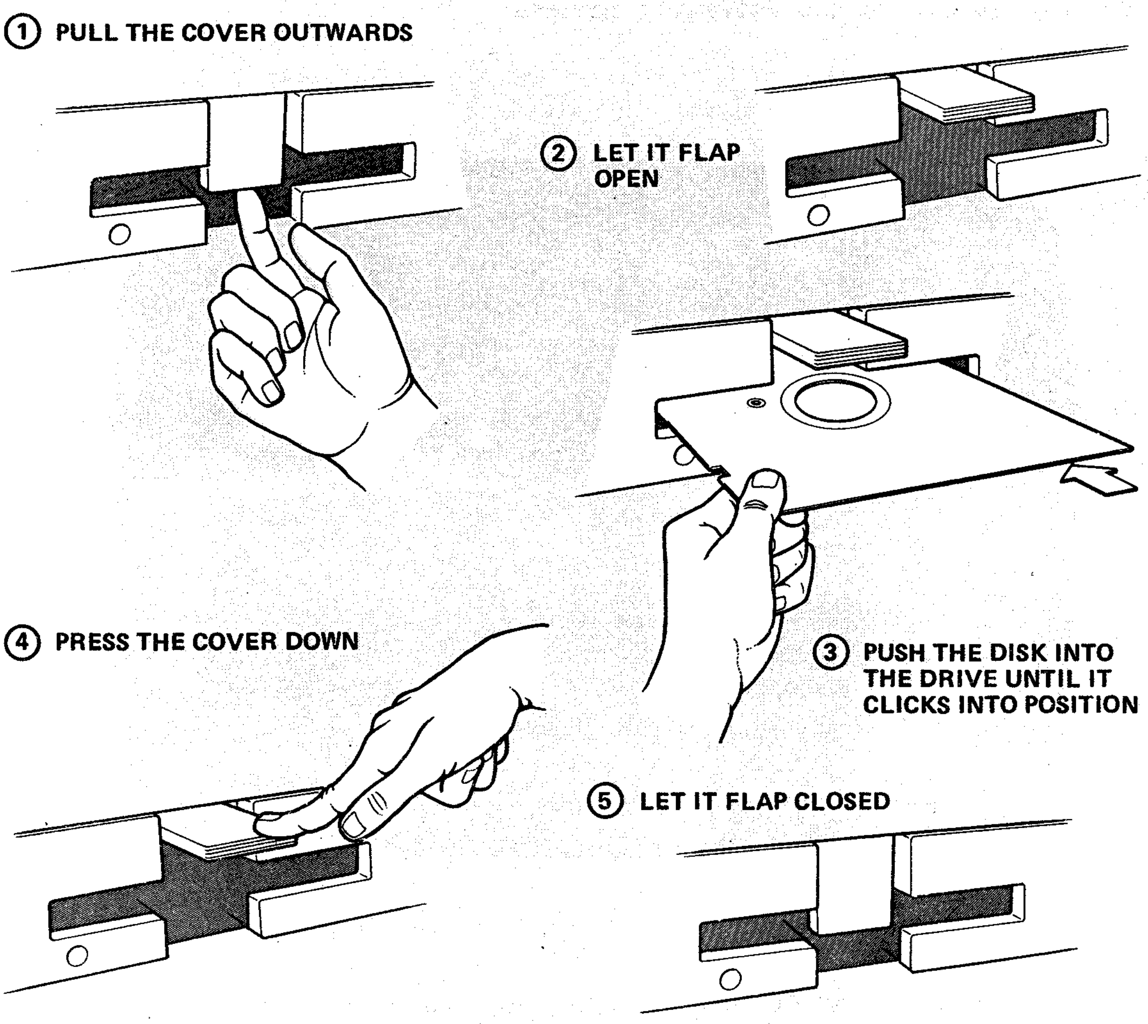

① PULL THE COVER OUTWARDS
② LET IT FLAP OPEN
③ PUSH THE DISK INTO THE DRIVE UNTIL IT CLICKS INTO POSITION
④ PRESS THE COVER DOWN
⑤ LET IT FLAP CLOSED

Figure 3-2 Loading and Unloading Disks

BASIC AND PCOS ENVIRONMENTS

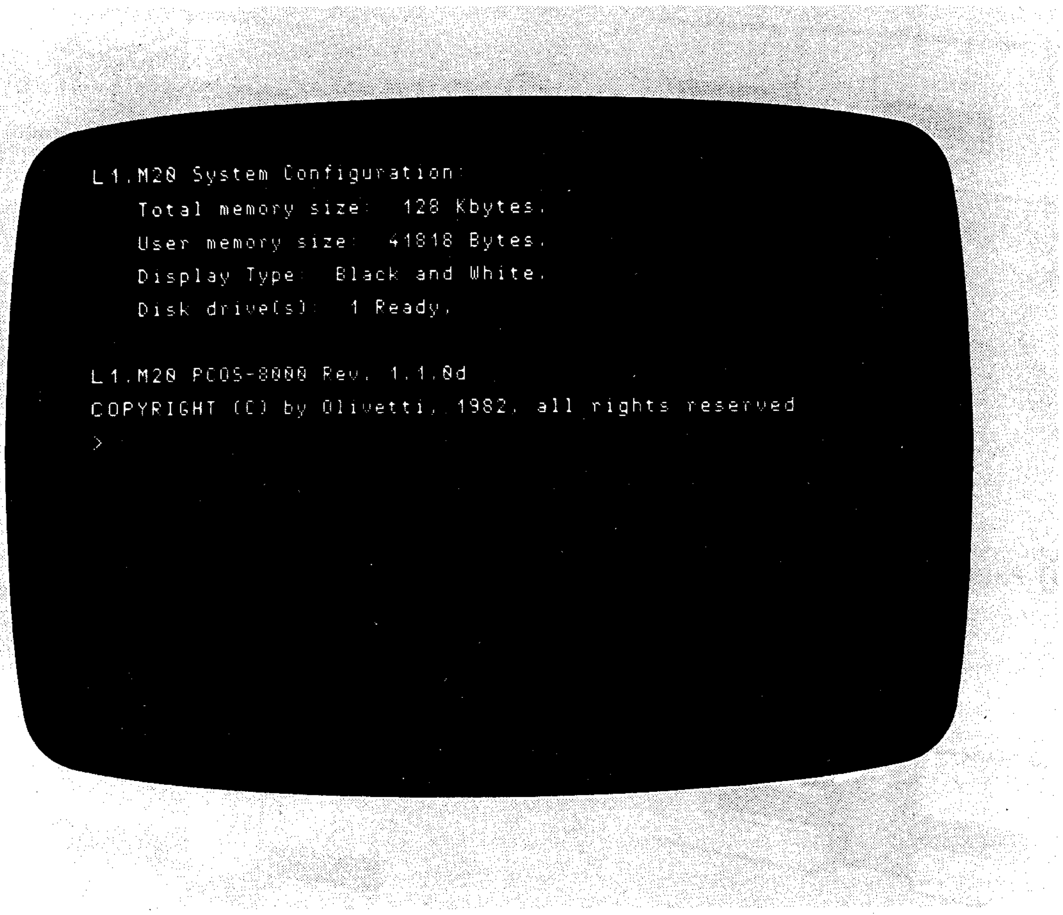
Before you can use Olisort, you have to load the BASIC Interpreter into the M20's memory.

In order to load the BASIC Interpreter into the M20 you have to exit from the PCOS environment and enter the BASIC environment. To do this you proceed as described below.

First power on the M20 as described earlier in this chapter. Then load the System Disk into the right hand drive. The correct method of loading disks was described earlier in this chapter.

Now press the CR key.

A message similar to the following appears:



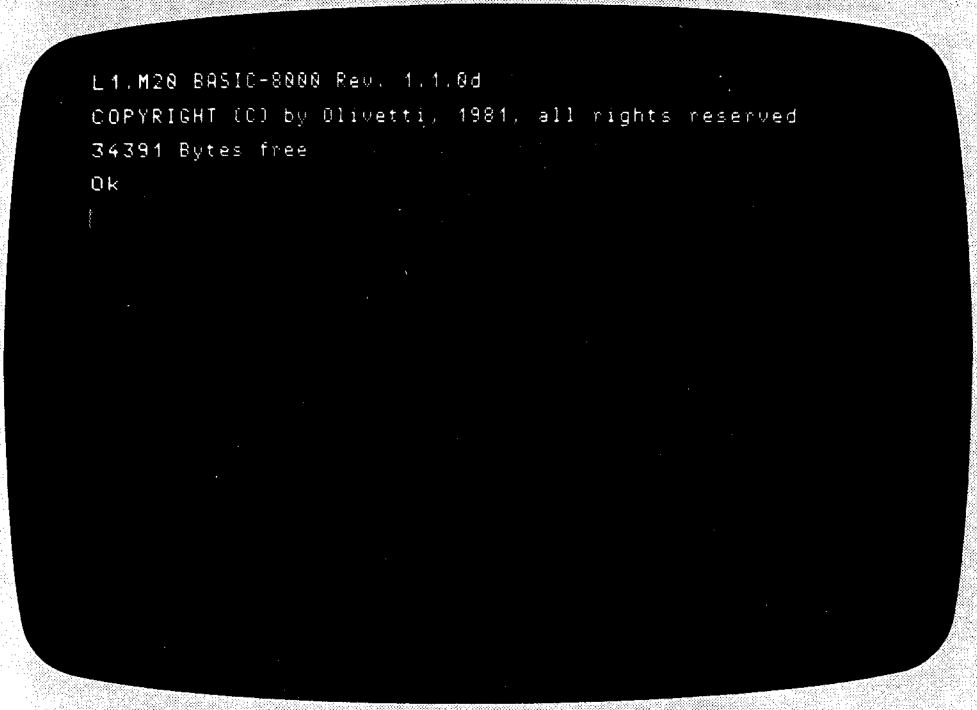
```
L1.M20 System Configuration:
   Total memory size:  128 Kbytes.
   User memory size:  41818 Bytes.
   Display Type:  Black and White.
   Disk drive(s):  1 Ready.

L1.M20 PCOS-8000 Rev. 1.1.0d
COPYRIGHT (C) by Olivetti, 1982, all rights reserved
>
```

This means that PCOS is loaded, and you could now work in the PCOS environment if you wanted to. The > symbol is what is known as the PCOS prompt. When you see the PCOS prompt followed by the cursor, you may enter PCOS commands, such as VFORMAT, VCOPY etc.

To leave the PCOS environment and enter BASIC, you type the following:

b a CR

A message similar to the following appears:



```
L1.M20 BASIC-8000 Rev. 1.1.0d
COPYRIGHT (C) by Olivetti, 1981, all rights reserved
34391 Bytes free
Ok
```

The BASIC Interpreter has now been loaded. The prompt Ok is what we call the BASIC prompt. Whenever you see the BASIC prompt followed by the cursor you can enter BASIC commands, such as LOAD, RUN, SAVE etc.

LISTING THE CONTENTS OF A DISK

Once you have entered the BASIC environment you can list the contents of any disks which are loaded. This is a useful facility to know about when you are using Olisort.

To list the contents of a disk, type the following:

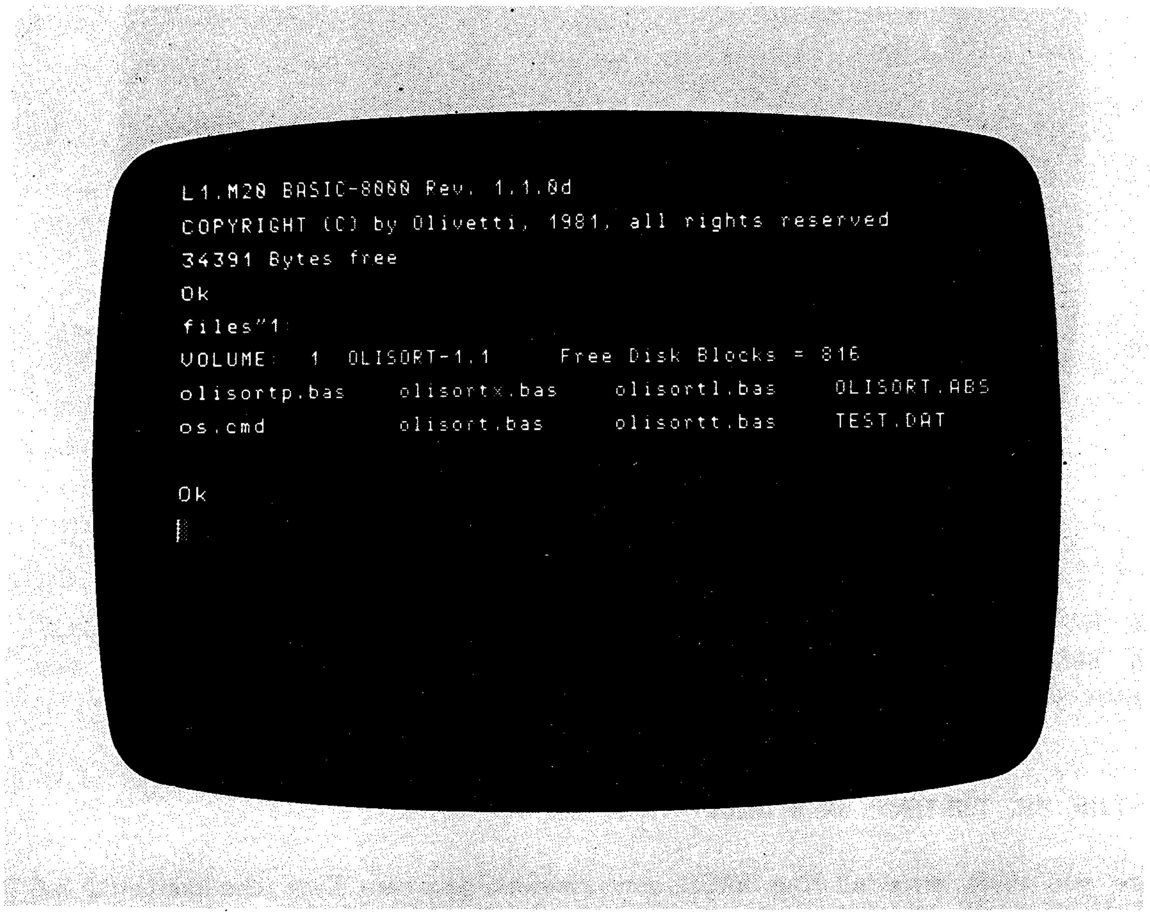
f i l e s " **d** : CR

Where **d** is the drive number housing the disk whose contents you wish to list, and is equal to Ø or 1.

The names of the files on the disk appear in alphabetical order on the screen. For example, if we put our OLISORT disk in the left hand drive and typed:

f **i** **l** **e** **s** **"** **1** **:** **CR**

the following would appear:



```
L1.M20 BASIC-8000 Rev. 1.1.0d
COPYRIGHT (C) by Olivetti, 1981, all rights reserved
34391 Bytes free
Ok
files"1:
VOLUME:  1  OLISORT-1.1      Free Disk Blocks = 816
olisortp.bas    olisortx.bas    olisortl.bas    OLISORT.ABS
os.cmd          olisort.bas     olisortt.bas    TEST.DAT

Ok
```

4. PARAMETERS

ABOUT THIS CHAPTER

Here we describe, in detail, all the parameters that you can pass to Olisort. We look closely at the four Olisort Modes and give examples of how to use them. Throughout the chapter abbreviations are used to signify parameters. These abbreviations are listed and explained in a fold-out page at the end of this chapter. The abbreviations are also found on the back of the Olisort Reference Card, and the contents of this chapter are summarized on the front.

This chapter is a tutorial on Olisort parameters which may also be used for reference. However, once you are familiar with Olisort you will probably find it more convenient to use the Olisort Reference Card.

CONTENTS

PARAMETER PASSING OVERVIEW

You can pass parameters to Olisort through a Parameter File on disk, or through a Command String that has been written into a BASIC program. The way you pass parameters is governed by the Olisort Mode you want to use.

You can create a Parameter File in the following ways:

- By using the utility olisortp (see Chapter 7).

- By using the utility olisortx and specifying Olisort Mode 3. (See Chapter 6 for olisortx, and this chapter for Olisort Mode 3.)

- By writing a Command String in a BASIC program and specifying Mode 3. (See Chapter 5 for writing Command Strings, and this chapter for Olisort Mode 3.)

You can create a Command String in the following ways:

- By coding the Command String into a BASIC program. (See Chapter 5.)

- By using the utility olisortx to interactively create a Command String and automatically run Olisort. (See Chapter 6.)

- By using the utility olisortx to create a Command String and save it on disk. This Command String can then be merged into a BASIC program. (See Chapter 6 for olisortx, and Chapter 5 for merging the Command String.)

SYNTAX AND OTHER NOTES

Some of the syntax we use to define Parameter String formats contain certain symbols which have a special meaning. These are as follows:

[ ] are used to enclose parameters which are optional

{ } are used to enclose a stack of possible values for a particular parameter

.... indicate a series of similar parameters

All other punctuation should appear in the Parameter String as it appears in the format.

Throughout this chapter and the remainder of the manual we refer to the OLISORT disk. By this we mean the working copy of the original Olisort disk, which you were advised to make in Chapter 1.

We also refer to the M20 disk drives by their drive numbers, that is:

- the right-hand drive is referred to as drive Ø

- the left-hand drive is referred to as drive 1.

PARAMETERS IN GENERAL

Parameters are passed to Olisort as a string of values separated by asterisk characters, '*'. This string of values is referred to as the Parameter String.

The Parameter String may include a reference to a Parameter File on disk, where the majority of the parameters will be contained.

Each Olisort Mode has its own Parameter String format, as defined under the appropriate heading in this chapter. Within Parameter String formats we use abbreviations to show individual parameters. The key to these abbreviations is found on the fold-out page at the end of this chapter, and on the Olisort Reference Card.

OLISORT MODE Ø

With Mode Ø you can perform a Sort, Select/Exclude, or Sort with Select/Exclude, based on the sort parameters passed through a Parameter File, previously saved on disk.

The format of the Parameter String is as follows:

O*sd*bd*p:parmfile*

Example

We have created a Parameter File, called PARAM1, on the disk in drive 1, and our OLISORT disk is in drive Ø. We want to put our BDUMP file onto the OLISORT disk, and perform a Sort as specified in the Parameter File.

If we coded our Command String into a BASIC program, it would look like this:

```
OLISORT.COM$="Ø*Ø*Ø*1:PARAM1*"
```

OLISORT MODE 1

With Mode 1 you can Merge two sorted files together, or Append one file onto the end of another, using a previously created Parameter File.

The Parameter String format for Mode 1 is as follows:

1*sd*bd*p:parmfile*i1:input1*i2:input2*o:output*

The contents of the Parameter File determine whether a Merge or Append will be performed:

- If no Sort Keys are specified in the Parameter File, i2:input2 is appended onto the end of i1:input1. The result is placed in o:output.

- If Sort Keys are specified, they must be the Sort Keys that were used to sort the two input files. In this case i1:input1 is merged with i2:input2, and the result is placed in o:output.

Examples

Merge: We have two sorted files called TRAN1 and TRAN2, on the same disk, and we want to merge them to produce a file called TRAN, also on the same disk. We have already created a parameter File called PARAM2, which is also on the same disk and contains the Sort Keys. This disk is loaded in drive 1.

The OLISORT disk is in drive Ø, and we want Olisort to write its BDUMP file there.

If we coded the Command String into a BASIC program, it would look like this:

```
OLISORT.COM$="1*Ø*Ø*1:PARAM2*1:TRAN1*1:TRAN2*1:TRAN*"
```

Append: We have two unsorted files called BILLS1 and BILLS2, on the same disk, and we want to append them to produce a file called BILLS, also on the same disk. We have already created a Parameter File called PARAM3 which is also on the same disk and contains no Sort Key information. This disk is in drive 1.

The OLISORT disk is in drive Ø and we want Olisort to write the BDUMP file on it.

If we coded the Command String into a BASIC program, it would look like this:

```
OLISORT.COM$="1*Ø*Ø*1:PARAM3*1:BILLS1*1:BILLS2*1:BILLS*"
```

OLISORT MODE 2

With Mode 2 you can perform a Sort, Select/Exclude, or a Sort and Select/Exclude. All the parameters are passed in a Command String in a BASIC program.

In the following examples SORT-KEY is:

fs*fl*ft*ad

and SELECT-EXCLUDE KEY is:

fs*fl*ft*leg*keyvalue

The Parameter String has the following formats:

- For a Sort

  **2*sd*bd*i1:input1*o:output*skp*rl*cw*co*w*SORT-KEY-1[*SORT-KEY-2....
  *SORT-KEY-1Ø]*AND*Ø***

- For a Select/Exclude

  2*sd*bd*i1:input1*o:output*skp*rl*cw*co*w*$\begin{Bmatrix}\text{AND}\\ \text{OR}\end{Bmatrix}$*SELECT-EXCLUDE-KEY-1

  [$\begin{Bmatrix}\text{AND}\\ \text{OR}\end{Bmatrix}$*SELECT-EXCLUDE-KEY-2....$\begin{Bmatrix}\text{AND}\\ \text{OR}\end{Bmatrix}$*SELECT-EXCLUDE-KEY-4]*Ø*

- For a Sort with Select/Exclude

  2*sd*bd*i1:input1*o:output*skp*rl*cw*w*SORT-KEY-1[*SORT-KEY-2....

  SORT-KEY-1Ø]*$\begin{Bmatrix}\text{AND}\\ \text{OR}\end{Bmatrix}$*SELECT-EXCLUDE-KEY-1 [*$\begin{Bmatrix}\text{AND}\\ \text{OR}\end{Bmatrix}$*SELECT-EXCLUDE-KEY-2....

  $\begin{Bmatrix}\text{AND}\\ \text{OR}\end{Bmatrix}$*SELECT-EXCLUDE-KEY-4]*Ø

Notes

Note that the list of Sort Keys always ends with the logical connector AND. Also the list of parameters always ends with a zero, which indicates the end of the list.

The AND is used in this context to indicate that there are no more Sort Keys and what follows is to be taken as a list of Select/Exclude Keys. The zero is taken to mean the end of the Select/Exclude Keys. Thus we have the following special cases:

- There are no Select/Exclude Keys. We then end the list of Sort Keys with AND*Ø*.

- There are no Sort Keys. We begin the list of Select/Exclude Keys with AND*.

Examples

To illustrate the use of Mode 2, we are going to use a small extract from the imaginary Stock File of a Power Tool Merchant. The extract is illustrated in Figure 4-1.

The record layout for each record in the file is as follows:

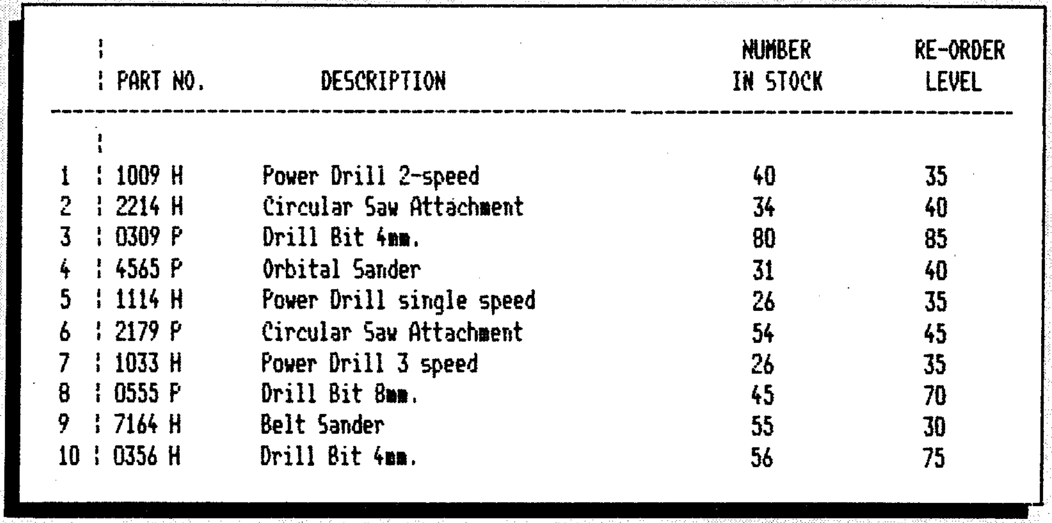
- PART NUMBER = 5 characters, the last character being an H, for Home/Hobby stock items, or a P for Professional stock items

- DESCRIPTION of up to 45 characters

- NUMBER IN STOCK = an Integer

- RE-ORDER LEVEL = an Integer which indicates how low the NUMBER IN STOCK can go before an order is placed with the supplier

The name of the file is STOCK.EXT.



```
   ¦                                             NUMBER      RE-ORDER
   ¦ PART NO.      DESCRIPTION                  IN STOCK      LEVEL
----------------------------------------------------------------------
   ¦
 1 ¦ 1009 H      Power Drill 2-speed               40           35
 2 ¦ 2214 H      Circular Saw Attachment           34           40
 3 ¦ 0309 P      Drill Bit 4mm.                    80           85
 4 ¦ 4565 P      Orbital Sander                    31           40
 5 ¦ 1114 H      Power Drill single speed          26           35
 6 ¦ 2179 P      Circular Saw Attachment           54           45
 7 ¦ 1033 H      Power Drill 3 speed               26           35
 8 ¦ 0555 P      Drill Bit 8mm.                    45           70
 9 ¦ 7164 H      Belt Sander                       55           30
10 ¦ 0356 H      Drill Bit 4mm.                    56           75
```

Figure 4-1 Stock File Extract

Sort: First we are going to sort the file into order of PART NUMBER within RE-ORDER LEVEL. To do this we need to use two Sort Keys, specifying them in the order of importance. That is to say, the first Sort Key will be for the RE-ORDER LEVEL, our major grouping, and the second Sort Key will be for PART NUMBER, our secondary grouping.

The Sort Keys are as follows:

- 53*2*I*A for RE-ORDER LEVEL

- 1*5*A*A* for PART NUMBER

We want our sorted output file, STOCK.SRT, to be written on the same disk as the input file. this disk will be loaded on drive 1.

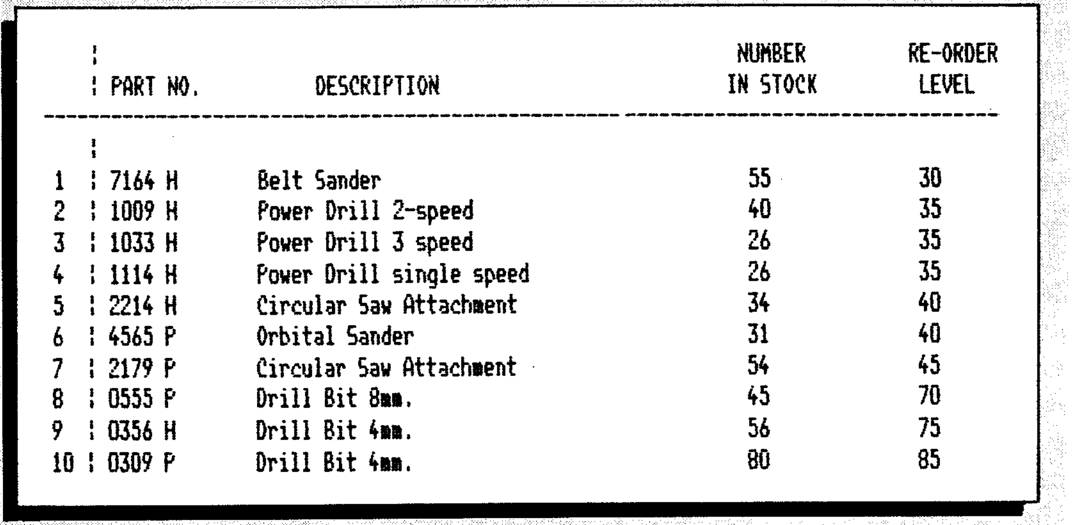
We will tell Olisort to write any Sort Work-Files on the disk in drive 1.

Our Olisort disk will be on drive Ø, and we want Olisort to write the BDUMP file there.

The Command String in a BASIC program would like like this:

```
OLISORT.COM$="2*Ø*Ø*1:STOCK.EXT*1:STOCK.SRT*Ø*54*N*N*1*53*2*I*A*1*5*I*A*
AND*O*"
```

After running the program which contained this Command String we would obtain the sorted output as illustrated in Figure 4-2.



```
    ¦                                                 NUMBER        RE-ORDER
    ¦ PART NO.        DESCRIPTION                    IN STOCK        LEVEL
-----------------------------------------------------------------------------
    ¦
 1  ¦ 7164 H      Belt Sander                           55            30
 2  ¦ 1009 H      Power Drill 2-speed                   40            35
 3  ¦ 1033 H      Power Drill 3 speed                   26            35
 4  ¦ 1114 H      Power Drill single speed              26            35
 5  ¦ 2214 H      Circular Saw Attachment               34            40
 6  ¦ 4565 P      Orbital Sander                        31            40
 7  ¦ 2179 P      Circular Saw Attachment               54            45
 8  ¦ 0555 P      Drill Bit 8mm.                        45            70
 9  ¦ 0356 H      Drill Bit 4mm.                        56            75
10  ¦ 0309 P      Drill Bit 4mm.                        80            85
```

Figure 4-2 Sorted Stock File

Select/Exclude: We want to select, from the Stock File, Drill Bits which are considered to be Professional Quality. That is stock items whose DESCRIPTION commences 'Drill Bit', and whose PART NUMBER ends with the letter P.

To do this we need two Select/Exclude Keys connected by AND. The order in which the Select/Exclude Keys appear in the Parameter String is unimportant.

We will tell Olisort to write its Sort Work-Files on the disk in drive 1.

We will write our output file to the disk which contains the input file. This disk will be on drive Ø and we will tell Olisort to write its BDUMP file there.

We would write our Command String as follows:

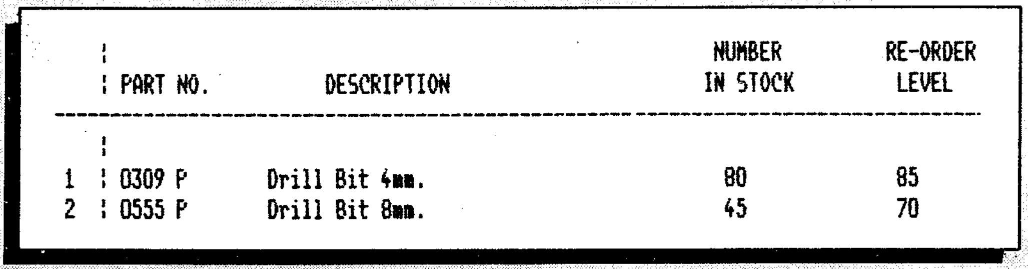
```
OLISORT.COM$="2*Ø*Ø*1:STOCK.EXT*1:STOCK.SE*O*54*N*N*1*Ø*AND*6*9*A*R*E*
drill=bit*AND*5*1*A*D*E*H*Ø*
```

Let's take a close look at this Command String:

- In the first Select/Exclude Key we have specified the key-value as "drill=bit". The equals sign is a Wild-Card Character, as discussed in Chapter 2. This means whatever character appears in the record in the corresponding position, it will be taken to be equal to the key-value. For instance, if the description of a Stock Item commenced 'Drill-Bit', it would still be considered to match the first Select/Exclude Key. Also, because we have specified the field type (ft) as A, the description in the record can commence with upper and/or lower case versions of the phrase Drill Bit, and it will still match the Select/Exclude Key. For instance, DRILL-BIT, drill bit, Drill*BIT and Drill Bit are all considered to match with the Select/Exclude Key.

- In the second Select/Exclude Key, we have specified D for the Delete/Retain Indicator. This means that we are excluding records which match this key from the output file. We are excluding all stock items whose PART NUMBER ends with the letter H, in order to select all those that end with the letter P. This is purely to show you how we specify exclude rather than select. We could have specified R for retain and put the key-value as P. For our Stock File extract, this would have exactly the same effect.

When we run the program which contains the Command String, we obtain the file illustrated in Figure 4-3.



```
   :                                              NUMBER      RE-ORDER
   : PART NO.        DESCRIPTION                 IN STOCK      LEVEL
-----------------------------------------------------------------------
   :
 1 : 0309 P       Drill Bit 4mm.                    80           85
 2 : 0555 P       Drill Bit 8mm.                    45           70
```

Figure 4-3 Select/Excluded Stock File

Sort and Select/Exclude: We want to select all stock items in the Home/Hobby category and sort them into order of STOCK NUMBER. To do this we need to create a Sort Key and a Select/Exclude Key.

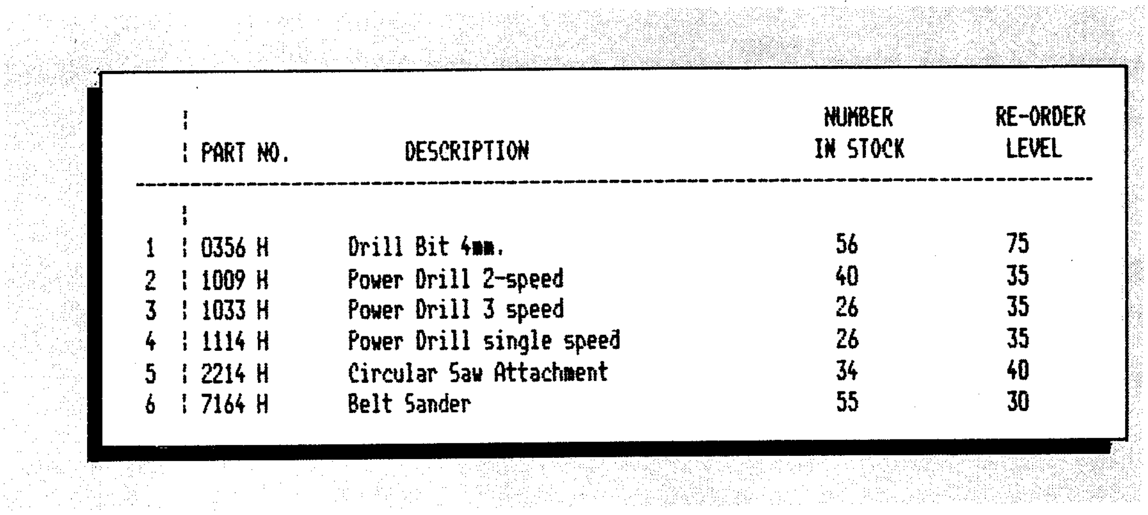
We are going to put the output file on the same disk as STOCK.EXT and call it STOCK/SSE. This disk will be in drive Ø and we will tell Olisort to write its BDUMP file there, and put any Sort Work-Files there.

The OLISORT disk will be loaded in drive 1.

The Command String looks like this:

```
OLISORT.COM$="2*1*Ø*Ø*:STOCK.EXT*Ø:STOCK.SSE*Ø*54*N*N*Ø*1*5*A*A*AND*5*1*
A*R*E*A*Ø*"
```

When we run the BASIC program which contains this Command String, we obtain the output illustrated in Figure 4-4.



```
     ¦                                              NUMBER       RE-ORDER
     ¦ PART NO.        DESCRIPTION                 IN STOCK        LEVEL
--------------------------------------------------------------------------
     ¦
  1  ¦ 0356 H      Drill Bit 4mm.                     56            75
  2  ¦ 1009 H      Power Drill 2-speed                40            35
  3  ¦ 1033 H      Power Drill 3 speed                26            35
  4  ¦ 1114 H      Power Drill single speed           26            35
  5  ¦ 2214 H      Circular Saw Attachment            34            40
  6  ¦ 7164 H      Belt Sander                        55            30
```

Figure 4-4 Sorted and Select/Excluded Stock File

OLISORT MODE 3

Mode 3 performs exactly the same way as Mode 2, except that a parameter File is created on disk. This Parameter File contains the parameters which were written into the Command String.

This is a useful facility if you want to Sort, Select/Exclude, or Sort and Select/Exclude, on more than one file, using the same keys. This was discussed in Chapter Two under the 'Select/Exclude Features' heading.

In the following examples SORT-KEY is:

fs*fl*ft*ad

and SELECT-EXCLUDE-KEY is:

fs*fl*ft*ad*key-value

The three Parameter String formats for Mode 3 are as follows:

- For a Sort

  3*sd*bd*p*parmfile*i1:input1*o:output*skp*rl*cw*co*w*SORT-KEY-1
  [*SORT-KEY-2....*SORT-KEY-1Ø]*AND*Ø*

- For a Select/Exclude

  3*sd*bd*p:parmfile*i1:input1*o:output*skp*rl*cw*co*w*$\begin{Bmatrix}\textbf{AND}\\\textbf{OR}\end{Bmatrix}$
  SELECT-EXCLUDE-KEY-1[**$\begin{Bmatrix}\textbf{AND}\\\textbf{OR}\end{Bmatrix}$SELECT-EXCLUDE-KEY-2....**$\begin{Bmatrix}\textbf{AND}\\\textbf{OR}\end{Bmatrix}$
  SELECT-EXCLUDE-KEY-4]*Ø

- For a Sort with Select/Exclude

  3*sd*bd*p:parmfile*i1:input1*o:output*skp*rl*cw*co*w*SORT-KEY-1
  [*SORT-KEY-2....SORT-KEY-1Ø]*$\begin{Bmatrix}\textbf{AND}\\\textbf{OR}\end{Bmatrix}$***SELECT-EXCLUDE-KEY-1**
  [*$\begin{Bmatrix}\textbf{AND}\\\textbf{OR}\end{Bmatrix}$***SELECT-EXCLUDE-KEY-2....***$\begin{Bmatrix}\textbf{AND}\\\textbf{OR}\end{Bmatrix}$***SELECT-EXCLUDE-KEY-4]*Ø***

Notes

Note that the list of Sort Keys always ends with the logical connector AND. Also the list of parameters always ends with a zero, which indicates the end of the list.

The AND is used in this context to indicate that there are no more Sort Keys and what follows is to be taken as a list of Select/Exclude Keys. The zero is taken to mean the end of the Select/Exclude Keys. Thus we have the following special cases:

- There are no Select/Exclude Keys. We then end the list of Sort Keys with AND*Ø*.

- There are no Sort Keys. We begin the list of Select/Exclude Keys with AND*.

Example

If we look back at the Sort example we gave in Mode 2, but this time we specify Mode 3, our Command String would look like this:

```
OLISORT.COM$="3*Ø*Ø*1*:PARAM4*1:STOCK.EXT*1:STOCK.SRT*Ø*54*N*N*1*53*2*I*
A*1*5*A*A*AND*Ø*
```

After running the BASIC program containing this Command String, we would obtain the same output file as before (see Figure 4-1). However, on the same disk we would find a Parameter File called PARAM4. THis Parameter File would contain the parameters from the Command String. We could use the Parameter File to sort STOCK.EXT again in Mode Ø. This time we would write the following Command String:

```
OLISORT.COM$="Ø*Ø*Ø*1:PARAM4"
```

ABBREVIATION	DESCRIPTION	VALUES	MEANING
		H I S	For Select/Exclude Keys. L means if all the characters in the key are less than the characters in the field, there is a match. E means if all the characters in the key are equal to the characters in the field, there is a match. G means if all the characters in the key are greater than the characters in the field, there is a match.
i1:input1	Input File 1	i1 = Ø or 1. input1 = any valid filename	Specifies the file where Olisort will write its output. For Sorts, o:output can be the same as i1:input1. In this case Input File 1 will be overwritten.
i2:input2	Input File 2	i2 = Ø or 1. input2 = any valid filename	For Modes Ø and 1, this specifies a Parameter File containing the input parameters for Olisort. For Mode 3, this specifies the parameter which will be created from Mode 3 Command String.
key-value	Select/Exclude Key value	any	Specifies in the length of each record in the Input File(s).
			Specifies the drive housing the OLISORT disk.
			Specifies the number of records in the file which will be skipped before Olisort begins Sorting, Merging etc.
			Specifies the drive housing the disk on which you want Olisort to write any Sort Work-Files.

5. PUTTING COMMAND STRINGS IN BASIC PROGRAMS

ABOUT THIS CHAPTER

Here we describe how you can put a Command String into a BASIC program. We do not describe the contents on the Command String, as that has been covered in Chapter 4.

It is assumed that, if you wish to put Command Strings into BASIC programs, you already have some knowledge of BASIC programming language.

You can learn how to write programs in BASIC using the BASIC Language Reference Guide, 398243Ø0.

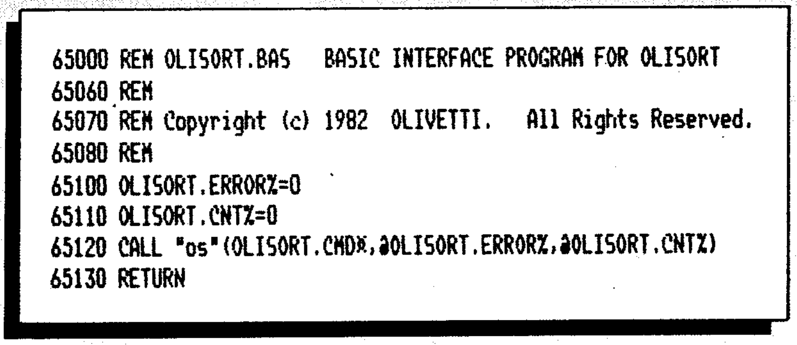
CONTENTS

PUTTING COMMAND STRINGS IN BASIC PROGRAMS

OVERVIEW

As we have said before, you can initiate Olisort via a BASIC program. this program may already exist, or you may want to write a new one. Either way you have to use the Olisort Interface Program in order to call Olisort.

The Olisort Interface Program resides on the OLISORT disk and is called olisort.bas. The program is illustrated in Figure 5-1.



```
65000 REM OLISORT.BAS   BASIC INTERFACE PROGRAM FOR OLISORT
65060 REM
65070 REM Copyright (c) 1982  OLIVETTI.   All Rights Reserved.
65080 REM
65100 OLISORT.ERROR%=0
65110 OLISORT.CNT%=0
65120 CALL "os"(OLISORT.CMD$,@OLISORT.ERROR%,@OLISORT.CNT%)
65130 RETURN
```

Figure 5-1 Olisort Interface Program

It is not a program in the true sense as it will not function until you have modified it. Once you have modified it, you can run it stand alone or merge it into a larger BASIC program.

Olisort always passes a status code to the calling program in a BASIC variable called OLISORT.ERROR%. This variable may be tested in the normal way. If an error is detected during processing, Olisort reloads the BASIC Interpreter and passes control back to the calling program. Therefore it makes sense to test OLISORT.ERROR% after each call to Olisort in your BASIC program. A full list of status codes is given in Appendix E.

Successful completion of Olisort is indicated by a status code of zero.

You may put as many calls to Olisort as you wish, in your BASIC program.

USING THE OLISORT INTERFACE PROGRAM

Before you can use the Olisort Interface Program you should enter into the BASIC environment, as described in Chapter 3. You should load the OLISORT disk on drive Ø.

In order to load the Olisort Interface Program into memory, type:

l o a d " Ø : o l i s o r t . b a s
" CR

When the BASIC prompt appears the program has been loaded. If you wished you could now list it on the screen by typing:

l i s t CR

or on the printer, by typing:

l l i s t CR

The program listing is illustrated in Figure 5-1.

As you can see, the subroutine which calls Olisort is at statement 65ØØØ. This means lines one through 64999 are available for your BASIC program.

To call Olisort you have to enter two lines of code. One will define the variable OLISORT.COM$, which contains the Command String, and the other will be GOSUB 65ØØØ, which calls Olisort.

You should also include another statement after the GOSUB 65ØØØ, which tests the contents of OLISORT.ERROR%, and proceeds accordingly.

Once you have modified the Olisort Interface Program to your satisfaction you can save it on the OLISORT disk by typing the following:

s a v e " Ø : **intprog** " CR

where **intprog** is the name you want to give to the modified version.

When the BASIC prompt appears, the program has been saved.

You could now run the program or merge it into another BASIC program. If you wanted to run the modified version as a stand along program, you would type:

r u n " Ø : **intprog** " CR

If you wanted to merge it into a BASIC program you would load the BASIC program into memory, and type:

m e r g e " Ø : **intprog** " CR

When the BASIC prompt appeared, the programs would have been merged.

If you are merging a modified version of the Olisort Interface Program into a BASIC program you should be aware of the following restrictions:

- The BASIC program must not have any line numbers greater than 64999

- the BASIC program should not contain any line numbers, which are the same as those you introduced into the Olisort Interface Program in order to modify it.

EXAMPLE

Let's return to the Stock File Extract we used in Chapter 4, which is illustrated in Figure 4-1.

We have a program which lists STOCK.EXT and resides on the same disk. The program is called STOCKLIST. What we want to do is insert a call to Olisort into STOCKLIST, so that we can obtain a listing in descending order of PART NUMBER.

To do this we proceed as described below.

First we enter into the BASIC environment, as described in Chapter 3. We load the OLISORT disk on drive Ø, and the disk containing STOCKLIST and STOCK.EXT on drive 1.

Next we type:

l o a d " Ø : o l i s o r t . b a s
" CR

When the BASIC prompt appears the program is loaded.

Our program STOCKLIST begins at line 2ØØ and ends at line 4ØØ. That means we can use line numbers from one through 199 for our statements to call Olisort. Let's say we choose to start at line 1ØØ. We enter the following BASIC statements:

```
100 OLISORT.COM$="2*Ø*Ø*1:STOCK.EXT*1:STOCK.SRT1*Ø*54*N*N*1*1*5*A*D*AND*
    Ø*"
11Ø GOSUB 65ØØ
12Ø IF OLISORT.ERROR%=Ø THEN GOTO 200 ELSE PRINT OLISORT.ERROR%
13Ø END
```

We have now modified the Olisort Interface Program and we want to save our modifed version on the OLISORT DISK, before merging it with STOCK-LIST. To do this we type:

s a v e " Ø : m y i n t p r o g " CR

When the BASIC prompt appears the program has been saved.

In order to merge we type:

m e r g e " 1 : S T O C K L I I S T " CR

When the BASIC prompt appears the programs have been merged.

Before running the program we want to save a copy of it on the same disk as STOCKLIST. So we type:

s a v e " 1 : L I S T A " CR

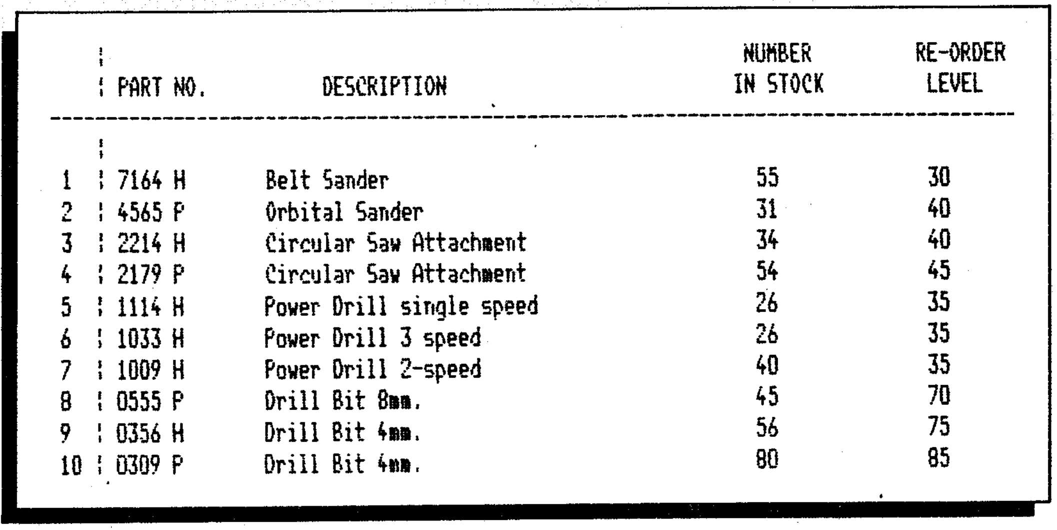
When the BASIC prompt appears we have a program called LISTA on the disk in drive 1, which will sort our Stock File into descending order of PART NUMBER and then print it. since the program is still in memory, as well as on the disk, we can run it by typing:

r u n CR

If we did that, the following message would appear on the screen:

```
Calling OLISORT

SORT INPUT AND SELECT PHASE
RECORDS READ OR SKIPPED=1Ø
RECORDS SELECTED=1Ø
OUTPUT PHASE
1Ø RECORDS WRITTEN TO OUTPUT FILE
```

The program would then produce the listing shown in Figure 5-2.



```
     ¦                                                   NUMBER       RE-ORDER
     ¦ PART NO.        DESCRIPTION                       IN STOCK      LEVEL
-------------------------------------------------------------------------------
     ¦
 1   ¦ 7164 H      Belt Sander                              55           30
 2   ¦ 4565 P      Orbital Sander                           31           40
 3   ¦ 2214 H      Circular Saw Attachment                  34           40
 4   ¦ 2179 P      Circular Saw Attachment                  54           45
 5   ¦ 1114 H      Power Drill single speed                 26           35
 6   ¦ 1033 H      Power Drill 3 speed                      26           35
 7   ¦ 1009 H      Power Drill 2-speed                      40           35
 8   ¦ 0555 P      Drill Bit 8mm.                           45           70
 9   ¦ 0356 H      Drill Bit 4mm.                           56           75
10   ¦ 0309 P      Drill Bit 4mm.                           80           85
```

Figure 5-2 Sorted Stock File from Program LISTA

6. THE olisortx UTILITY

ABOUT THIS CHAPTER

Here we tell you how to use the utility olisortx that allows you to create Command Strings, interactively. For lengthy Command Strings with several key definitions, this appreciably reduces the chance of making keying errors.

The olisortx utility program is intended to be self-explanatory. This chapter is intended to clarify the use of olisortx for inexperienced users, and for this reason, depending on your previous experience, you may find it gives you more information than you need.

CONTENTS

THE olisortx UTILITY

OVERVIEW

The olisortx utility resides on the OLISORT disk with the name olisortx.bas.

There are three principal uses for olisortx, as follows:

- To Sort, Select/Exclude, Sort and Select/Exclude, Merge, or Append on an existing file. This is the ideal way to generate Mode 2 command Strings if you want to use Olisort to process a file only once (option S).

- To create a Command String and then save it on disk in order to merge it into a BASIC program (option C).

- To create a Command String which will be displayed on the screen, so you can copy the correct Command String form, in order to code it into a BASIC program (option B).

STANDARD ERROR MESSAGE AND RESPONSE ENDING

The olisortx utility lets you build a Command String via a series of prompts that appear on the screen. Your response to each of these prompts must be terminated by pressing the CR key.

While making responses the following error message may appear:

≫Invalid Response. Please re-enter.

You should then re-key the response, ensuring that it is within the limits defined in the prompt.

RUNNING olisortx

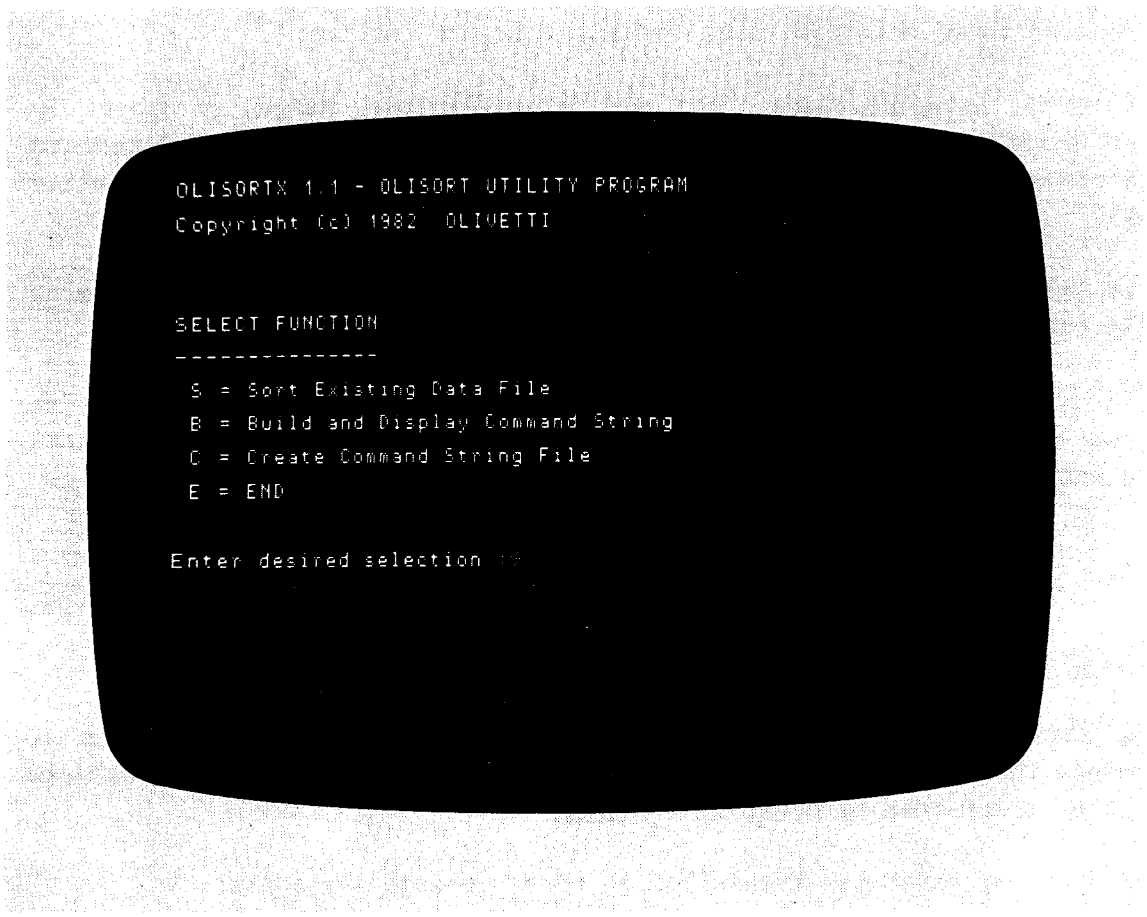
Using the OLISORT disk in drive Ø, enter the BASIC environment, as described in Chapter 3.

When the BASIC prompt appears, type the following:

r u n " Ø : o l i s o r t x . b a s
" CR

OLISORTX OPTION MENU

The following then appears on the screen:



```
OLISORTX 1.1 - OLISORT UTILITY PROGRAM
Copyright (c) 1982  OLIVETTI


SELECT FUNCTION
---------------
 S = Sort Existing Data File
 B = Build and Display Command String
 C = Create Command String File
 E = END

Enter desired selection
```

You now type **s**, **b**, **c**, or **e**, depending on what you want to do.

If You Chose Option E

The following message will appear:

Ending OLISORTX

and the program will terminate. The BASIC prompt appears.

If You Chose Any Other Option

The following message appears:

Enter Sort Mode (Ø,1,2,3):

You must now type in **Ø**, **1**, **2**, or **3**, depending on the Mode you want to use. You will then be prompted to enter the parameters which make up the Command String.

Table 6-1 shows you the prompts that may appear for the various Modes, and the corresponding parameters that must be entered. The parameters are shown in their abbreviated form, as used on the fold-out page at the end of Chapter 4, and the Olisort Reference Card.

OLISORTX PROMPT	MODES FOR WHICH PROMPT APPEARS	PARAMETER TO BE ENTERED
Enter drive I.D. for sort program (Ø,1,1Ø,11, 12)	ALL	sd
Enter drive I.D. for memory storage file (Ø,1, 1Ø,11,12)	ALL	bd
Enter drive I.D. for parameter file (Ø,1,1Ø, 11,12)	Ø, 1, and 3	p
Enter parameter file name (Ø,1,1Ø,11,12)	Ø, 1, and 3	parmfile
Enter drive I.D. for input file (Ø,1,1Ø,11,12)	2 and 3	i1
Enter input file name:	2 and 3	input1
Enter drive I.D. for Input File 1 (Ø,1,1Ø,11, 12)	1	i1
Enter Input 1 file name (Ø,1,1Ø,11,12)	1	input1
Enter drive I.D. for input file 2 (Ø,1,1Ø,11, 12)	1	i2
Enter input 2 file name:	1	input2
Enter drive for output file (Ø,1,1Ø,11,12)	1, 2 and 3	o
Enter output file name:	1, 2 and 3	output
Enter number of records to skip (Ø-32767):	2 and 3	skp

Enter record length:	2 and 3	rl
Change work disk? (Y/N):	2 and 3	cw
Change output disk? (Y/N):	2 and 3	co
Enter drive I.D. for work file (Ø-1Ø):	2 and 3	w

Table 6-1 Parameters for olisortx

ENTERING SORT KEYS

If you chose Mode 2 or Mode 3, the following prompt appears:

Enter number of sort keys (Ø-1Ø):

You have to key in an integer between Ø and 1Ø, which specifies the number of Sort Keys you want to create. If you enter Ø that means you don't want to enter any Sort Keys, so proceed to the heading 'Entering Select/Exclude Keys'.

If you entered anything other than Ø you will be prompted to enter each Sort Key individually, via the following message:

Enter parameters for sort key **n**

Where **n** is the number of the Sort Key to be entered next.

The prompts for each Sort Key and the parameters to be entered are illustrated in Table 6-2. The parameters in the table are shown in their abbreviated forms as listed on the fold-out page in Chapter 4, and on the Olisort Reference Card.

SORT KEY PROMPT	PARAMETER	NOTES
Starting position (1-rl)	fs	Where rl = the value you were prompted to enter earlier.
Key length (1-n)	fl	Where n = rl less fs. If you enter a value greater than n, the following error message appears: >>Key extends past end of record You have to key in a value less than n.
Field Type: (A=Alpha H=Hex I=Int S=Sngl.Prec D=Dbl.Prec)	ft	
Sort in Ascending or Descending order (A/D)	ad	

Table 6-2 Sort Key Prompts

After you have entered the last Sort Key you will be prompted to enter any Select/Exclude Keys.

ENTERING SELECT/EXCLUDE KEYS

The following message appears:

Enter the number of select keys (Ø-4);

You must now enter an integer between one and four, which specifies the number of Select/Exclude Keys that you want to create. If you enter Ø you have no Select/Exclude Keys to create and you can proceed to the heading, 'Displaying the Command String'.

If you enter any other number you will be prompted to enter each Select/ Exclude Key, individually.

The prompts for each Select/Exclude Key and the parameters you enter are shown in Table 6-3. The parameters in the table are given in their abbreviated forms as listed on the fold-out page in Chapter 4, and on the Olisort Reference Card.

SELECT/EXCLUDE KEY PROMPT	PARAMETER	NOTES
Starting position (1-rl)	fs	Where rl = the value you were prompted to enter, earlier.
Key length (1-n)	fl	Where n = rl less fs. If you enter a value greater than n, the following error message appears: >>Key extends past end of record You have to key in a value less than n.
Field Type: (A=Alpha H=Hex I=Int S=Sngl.Prec D=Dbl.Prec)	ft	
Retain or Delete (R/D)	rd	
Comparison type: (L =less than E=equal to G=greater than)	leg	
Select key value:	key value	

Table 6-3 Select/Exclude Key Prompts

After you have entered the last Select/Exclude Key, the Command String you have created will be displayed, as described under the following heading.

DISPLAYING THE COMMAND STRING

If You Chose Option C

the following message appears:

Enter drive I.D. for Command String File (Ø-1):

Enter the drive number that houses the disk on which you want olisortx to write your Command String.

The following message appears:

Enter Command String File name:

Enter any valid filename. This filename will be assigned to the Command String file on the disk.

The following message appears:

Enter line number for Command String:

Enter any line number from one to 64999. A file will be created on disk which contains one line of BASIC code. The line will have the number which you have just entered. You can then merge this file into the Olisort Interface Program, using the method described in Chapter Five.

Whether You Chose Option C Or Not

The following appears:

Sort Parameter string =
Command String

where the **Command String** is the Command String you have just created, exactly as it would appear in a BASIC program.

If you chose Option C, the Command String displayed will include the line number you assigned to it, and the following message will appear:

Olisort Parameter String successfully written

Finally, for all options except S, the following message appears:

Hit return to continue

If you wanted to, you could now make a note of the displayed Command String. When you are ready to proceed, press CR. If you chose any option other than S, the program will proceed as described under the heading 'OLISORTX OPTION MENU'.

If You Chose Option S

The following message appears:

Enter s to sort or e to end

If you enter **e** the program proceeds as described under the heading 'OLISORTX OPTION MENU'.

If you wish to sort, load the disk containing your input files and input Parameter File (for Modes Ø and 1) onto the drive specified. Check that the OLISORT disk is loaded onto the specified drive. Then press **s** and Olisort begins processing, as specified in the Command String that you have just created.

If an erroneous status code is detected during processing, it is reported to you on the screen and olisortx ends. The BASIC prompt then re-appears. For a full list of status codes and their meanings, refer to Appendix E.

If no errors are detected, Olisort produces the following messages:

```
Calling OLISORT

SORT INPUT AND SELECT PHASE
RECORDS READ OR SKIPPED = n
RECORDS SELECTED = m
OUTPUT PHASE
m RECORDS WRITTEN TO FILE

OLISORT successfully completed
Hit return to continue
```

Where **n** and **m** are integers

Your processed output file now exists as specified in the parameter o:output.

When you are ready to continue press **CR**, and the program will proceed as described under the 'OLISORTX OPTION MENU' heading.

7. THE olisortp UTILITY

ABOUT THIS CHAPTER

Here we tell you how to use the utility olisortp that allows you to create Parameter Files, interactively.

The olisortp utility program is intended to be self-explanatory. This chapter is intended to clarify the usage of olisortp for inexperienced users, and for this reason, depending on your previous experience, you may find it gives you more information than you need.

CONTENTS

THE OLISORTP UTILITY

OVERVIEW

The olisortp utility resides on the OLISORT disk with the name olisortp.bas.

There are two principal uses for olisortp, as follows:

- To create a Parameter file and then save it on disk

- To display a Parameter File.

STANDARD ERROR MESSAGE AND RESPONSE ENDING

The olisortp utility lets you build a Parameter File via a series of prompts that appear on the screen. Your response to each of these prompts must be terminated by pressing the CR key.

While making responses the following error message may appear:

```
  Invalid Response. Please re-enter.
```

You should then re-key the response, ensuring that it is within the limits defined in the prompt.

RUNNING olisortp

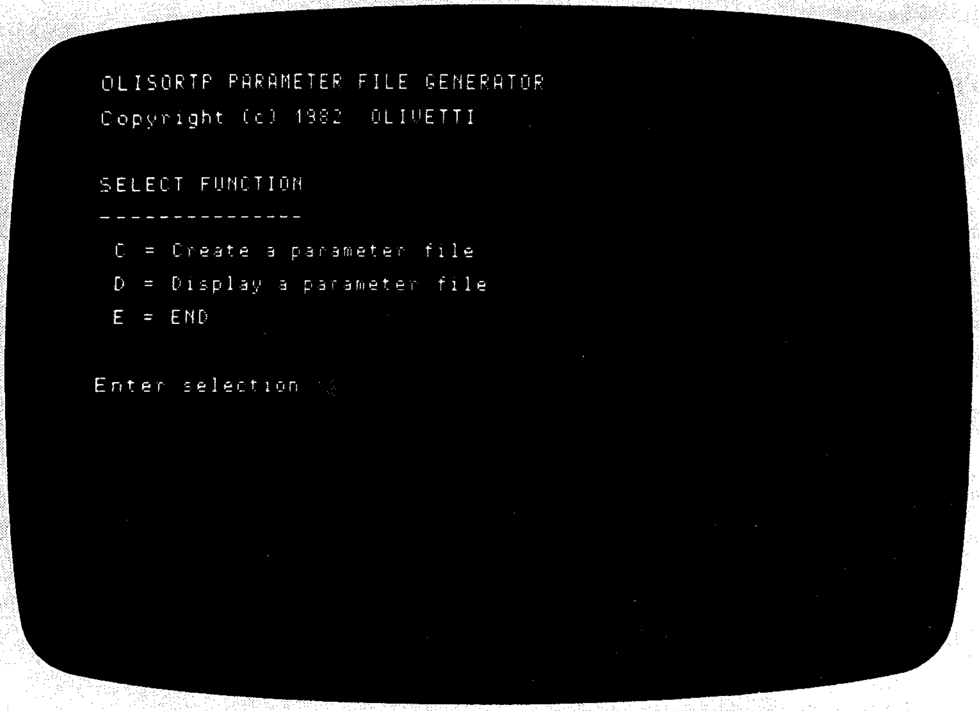
Using the OLISORT disk in drive Ø, enter the BASIC environment, as described in Chapter 3.

When the BASIC prompt appears, type the following:

r u n " Ø : o l i s o r t p . b a s " CR

OLISORTP OPTION MENU

The following then appears on the screen:



```
OLISORTP PARAMETER FILE GENERATOR
Copyright (c) 1982  OLIVETTI

SELECT FUNCTION
---------------
 C = Create a parameter file
 D = Display a parameter file
 E = END

Enter selection ?
```

You now type c, d, or e, depending on what you want to do.

If You Chose Option E

The following message will appear:

Ending OLISORTP

and the program will terminate. The BASIC prompt appears.

If You Chose Any Other Option

The following message appears:

Enter drive I.D. of parameter file (Ø-1):

Enter the number of the drive that houses the disk containing the Parameter File you want to display, or on which you want to write the Parameter File you are going to create. The following message appears:

Enter parameter file name

Enter the name of the Parameter File that you want to display, or to create.

If You Chose Option D

The following error message may appear:

File not found

If so, check that you have the correct disk in the correct drive. If you want to check the contents of a disk, you can use the method described in Chapter 4, under the heading 'Listing the Contents of a Disk'.

If the Parameter File is found it will be displayed on the screen followed by the message:

Hit return to continue

When you are ready, press CR. the program proceeds as described under the heading 'OLISORTP OPTION MENU'.

If You Chose Option C

You will be prompted to enter a list of parameters. In Table 7-1 we give you a list of the prompts and the corresponding parameters. The parameters are shown in their abbreviated form, as used on the fold-out page at the end of Chapter 4, and the Olisort Reference Card.

OLISORTP PROMPT	PARAMETER TO BE ENTERED
Enter drive I.D. for input file (Ø,1,1Ø,11,12)	i1
Enter input file name:	input1
Enter drive for output file (Ø,1,1Ø,11,12)	o
Enter output file name:	output
Enter number of records to skip (Ø-32767)	skp

▶

Enter record length:	rl
Change work disk? (Y/N):	cw
Change output disk? (Y/N)	co
Enter drive I.D. for work file (Ø,1,1Ø,11,12)	w

Table 7-1 Parameters for olisortp

ENTERING SORT KEYS

The following prompt appears:

Enter number of sort keys (Ø-1Ø):

You have to key in an integer between Ø and 1Ø, which specifies the number of Sort Keys you want to create. If you enter Ø that means you don't want to enter any Sort Keys, so proceed to the heading 'Entering Select/Exclude Keys'.

If you entered anything other than Ø you will be prompted to enter each Sort Key individually, via the following message:

Enter parameters for sort key **b**

Where **n** is the number of the Sort Key to be entered next.

The prompts for each Sort Key and the parameters to be entered are illustrated in Table 7-2. The parameters in the table are shown in their abbreviated forms as listed on the fold-out page in Chapter Four, and on the Olisort Reference Card.

SORT KEY PROMPT	PARAMETER	NOTES
Starting position (1-rl)	fs	Where rl = the value you were prompted to enter, earlier.
Key length (1-n)	fl	Where n = rl less fs. If you enter a value greater than n, the following error message appears: >>Key extends past end of record You have to key in a value less than n.
Field Type: (A=Alpha H=Hex I=Int S=Sngl.Prec D=Dbl.Prec)	ft	
Sort in Ascending or Descending order (A/D)	ad	

Table 7-2 Sort Key Prompts

After you have entered the last Sort Key you will be prompted to enter any Select/Exclude Keys.

ENTERING SELECT/EXCLUDE KEYS

The following message appears:

Enter the number of select keys (Ø-4);

You must now enter an integer between 1 and 4, which specifies the number of Select/Exclude Keys that you want to create. If you enter Ø you have no Select/Exclude Keys to create and you can proceed to the heading, 'DISPLAYING THE PARAMETER FILE'.

If you enter any other number you will be prompted to enter each Select/Exclude Key, individually.

The prompts for each Select/Exclude Key and the parameters you enter are shown in Table 7-3. The parameters in the table are given in their abbreviated forms as listed on the fold-out page in Chapter 4, and on the Olisort Reference Card.

SELECT/EXCLUDE KEY PROMPT	PARAMETER	NOTES
Starting position (1-rl)	fs	Where rl = the value you were prompted to enter, earlier.
Key length (1-n)	fl	Where n = rl less fs. If you enter a value greater than n, the following error message appears: >>Key extends past end of record You have to key in a value less than n.
Field Type: (A=Alpha H=Hex I=Int S=Sngl.Prec D=Dbl.Prec)	ft	
Retain or Delete (R/D)	rd	
Comparison type: (L=less than E=equal to G=greater than)	leg	
Select key value:	key value	

Table 7-3 Select/Exclude Key Prompts

After you have entered the last Select/Exclude Key, the Parameter File you have created will be displayed, as described under the following heading.

WRITING THE PARAMETER FILE

The Parameter File is then displayed on the screen, followed by the message:

```
Parameter File successfully created
Hit return to continue
```

You now have a Parameter File on disk which you can refer to from a Command String.

When you are ready press the CR key.

The program will proceed as described under the heading 'OLISORTP OPTION MENU'.

A. WRITE PROTECTION OF DISKS

ABOUT THIS APPENDIX

Here we tell you how to physically Write Protect your disks.

WRITE PROTECTION OF DISKS

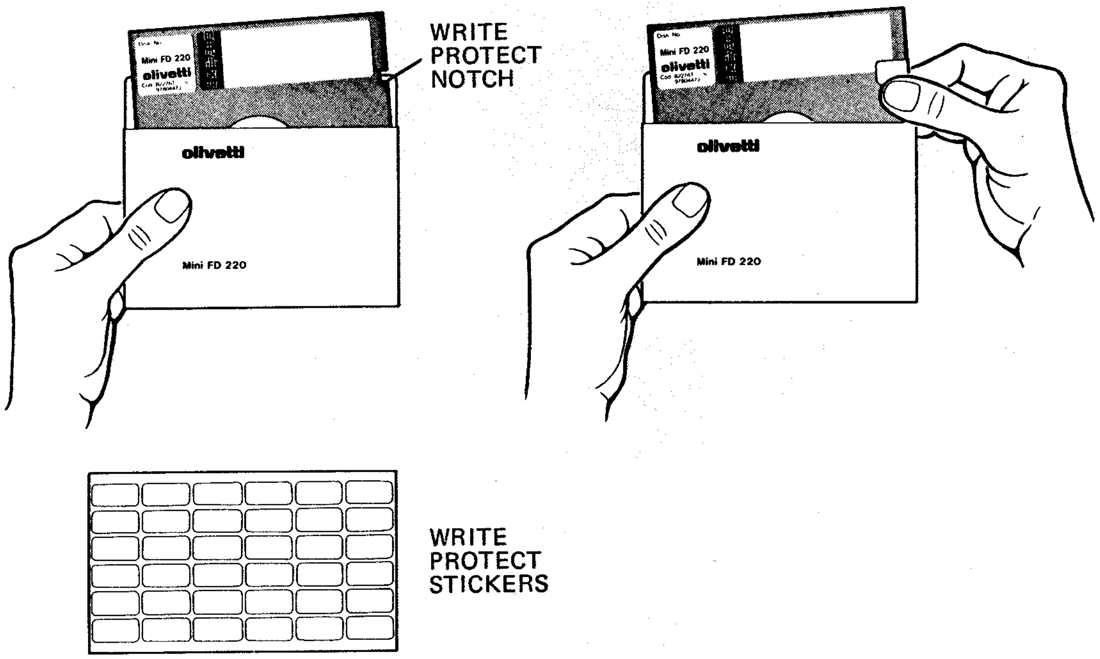


Figure A-1 Write Protection Method

If you wish to secure any of the information on the disks associated with your application, you can physically write protect them. Write protection ensures that the information on a disk cannot be overwritten by accident.

The method of write protecting a disk is illustrated in Figure A-1. You merely peel a sticker off the sheet supplied with each carton of new disks, and wrap it around the write protect notch on the edge of the disk.

An attempt to write information to a write protected disk will result in an error.

If you decide that you do want to add information to a write protected disk, you can remove the write protection by peeling off the sticker.

B. BACKUPS

ABOUT THIS APPENDIX

Here we tell you how to make a backup copy of a disk, and how to recover from the original disk in the event of a machine fault or power failure.

CONTENTS

BACKUPS

It is always advisable to keep one, or more, backup copies of any disks that contain important information. It is strongly advised that you keep copies of the original disks that come with your application package.

There are two procedures described in this appendix. One for making a backup copy, and one for recovering from a backup copy, in case an unforeseeable error occurs while one of your disks is in use. Unforeseeable errors include power failures, physical damage to disks, etc.

To make a backup copy you always need a formatted disk. To recover from a backup copy you sometimes need a formatted disk. A formatted disk may be one of the following:

- a used disk which contains information that you no longer have any use for, and has been formatted via the PCOS command VFORMAT (See Appendix C).

- an Olivetti pre-formatted disk.

- a new Olivetti disk that has been formatted via the PCOS command VFORMAT. (See Appendix C.)

MAKING A BACKUP COPY

1. Power on the M20, as described in Chapter 3, 'Power On/Off' section.

2. Load a PCOS system disk (as illustrated in Figure 1-1) onto the right-hand drive, and the disk you want to copy onto the left-hand drive. Ensure that both disks are write protected, as described in Appendix A.

3. Enter:

   v **c** **SPACE** **1** **:** **SPACE** **Ø** **:** **CR**

4. The following message appears:

   ```
   Warning - vcopy deletes all files. Copy disk? (y/n)
   ```

   If any other message appears, you have probably made a keying error. In this case, return to Step 3.

5. Remove the system disk from the right hand-drive and insert the disk you are copying on to. This disk should NOT be write protected. Enter:

   y CR

6. The disk in the left-hand drive is copied onto the one in the right-hand drive. This takes a couple of minutes. When the copying procedure is complete the PCOS prompt > appears. You may then remove your disks.

RECOVERING FROM THE ORIGINAL

There are two different situations in which you may need to recover from the original. They are as follows:

- the disk you are recovering has been physically damaged.

- the disk you are recovering is not physically damaged, but the information on it has been corrupted.

In the first case you have to throw the damaged disk away and obtain a formatted disk to recover onto, as described earlier in this appendix. In the second case, you can recover onto the corrupted disk.

In either case the procedure is the same as that described for making a backup copy. You are recovering from the original copy, by making a new backup copy of the original.

C. FORMATTING DISKS

ABOUT THIS APPENDIX

Here we tell you how to format a new disk, or an old disk that you want to re-cycle.

FORMATTING DISKS

Before you can use a new unformatted disk, or in order to re-cycle an old disk, you have to format it. You can do this via the procedure given here.

First get the System Disk, as illustrated in Figure 1-1. It should already be write protected. If it isn't write protect it now, as described in Appendix A.

Power on the M20 as described in Chapter 3, under the 'Power On/Off' heading. Then load the System disk into the right-hand drive. The correct method of loading disks is described in Chapter 3, under the 'Loading Disks' heading. Now press the **CR** key.

When the PCOS prompt > appears, proceed as described below.

Load the disk you want to format onto the right-hand drive. This disk must not be write protected. Now enter the following:

v **f** **SPACE** **1** **:** **CR**

A message similar to the following one appears:

```
Volume Format Rev 1.1.Ø.d
Warning: VFORMAT deletes all files. Format disk? (y/n)
```

This is to ensure that you do not accidentally destroy the wrong disk. At this point you take the disk out of the left-hand drive and check. When you are sure that you have loaded the correct disk, enter:

y **CR**

If the disk you are formatting is a new one, the formatting begins immediately and ends with the message:

```
Formatting Completed
```

If you are re-formatting an old disk, one or both of the following messages may appear:

```
Diskette appears password protected. Format disk? (y/n)
Diskette appears OPE formatted. Format disk anyway? (y/n)
```

If you are sure that the disk in the right-hand drive is the one you want to format, enter:

y **CR**

as the response to both messages.

The formatting procedure will begin as described above.

when the disk has been formatted, you may remove it from the right-hand drive and use it to make backups, save files, etc.

D. PCOS AND BASIC ERROR MESSAGES

ABOUT THIS APPENDIX

Here we give you a full list of the error messages which may be produced while you are using the M20. This is to allow you the possibility of correcting errors which occur without making reference to the BASIC or PCOS manuals. If you cannot resolve a problem which occurs by using this appendix, you will have to seek advice from a BASIC programmer.

PCOS AND BASIC ERROR MESSAGES

Errors returned from the BASIC Interpreter are not displayed with their error number (only the description is displayed). Errors returned from PCOS are displayed with both the error number and the descriptive label, e.g.:

```
sf %n,pr24ØØ?15,12

error 9Ø in parameter 2
```

These error code lists occasionally show comments explaining the conditions which give rise to an error and/or indicating to the user a possible course of action on appearance of the error message and/or error code.

There are in all 256 BASIC error codes and 127 PCOS error codes, however neither PCOS nor BASIC use all the codes available. In BASIC, unused error codes display the message "Unprintable error". In PCOS, unused error codes display the error number without any message. Unused error codes are not listed in this appendix.

BASIC ERRORS		
ERROR CODE	MESSAGE	COMMENT
1	Next without FOR	A NEXT statement has been encountered without a matching FOR
2	Syntax error	A line has been encountered which includes an incorrect sequence of characters (mis-spelt keyword, incorrect punctuation etc.)
3	RETURN without GOSUB	A RETURN has been encountered for which there is no previous unmatched GOSUB statement
4	Out of data	A READ statement has been executed when there are no DATA statements with unread data remaining in the program

5	Illegal function call	A parameter that is out of range has been passed to a numeric or a string function Such an error may occur when: a. An array subscript is either negative or too big b. A log function is assigned a negative or a null argument c. The SQR function is assigned a negative value d. A negative number has an exponent which is not an integer e. A USR function has been called without having established the initial address f. An incorrect argument has been made in one of the following functions: MID$, LEFT$, RIGHT$, TAB, SPC, STRING$, SPACES$, INSTR, or ON...GOTO
6	Overflow	The result of a calculation is too large to be represented in BASIC's number format Note: With underflow, the result is taken as Ø, and execution continues without indication of an error
7	Out of memory (also used in PCOS)	A program is too big; or has too many loops, GOSUBS, variables; or has expressions too complicated to evaluate
8	Undefined line	A line reference is to a non-existent line from a GOTO, GOSUB, IF..THEN..ELSE or DELETE

9	Subscript out of range	An array element has been referred to either with a subscript that is outside the dimensions of the array or with the wrong number of subscripts
1Ø	Duplicate definition	Two DIM statements have been given for the same array, or a DIM statement has also been applied to an array after the default dimension of 1Ø was previously established for that array
11	Division by zero	A division by zero has been encountered or the value zero has been raised to a negative power. In the former case the result is machine infinity (with the appropriate sign) and in the latter case the result is positive machine infinity
12	Illegal direct	A statement which is invalid in immediate (direct) mode has been entered as an immediate command
13	Type mismatch (also used in PCOS)	A string variable name has been assigned a numeric value or vice versa; a function that expects a numeric argument has been given a string argument or vice versa
14	Out of string space	String variables have caused BASIC to exceed the amount of free user memory remaining. (BASIC will allocate space dynamically until it runs out of memory)
15	String too long	An attempt has been made to create a string more than 255 characters long

16	String formula too complex	A string expression is too long or too complex to be processed. It should be broken into smaller expressions
17	Can't continue	An attempt has been made to continue a program that cannot be continued: as it is halted due to an error; was modified during a break in execution; or does not exist in user memory
18	Undefined user function	A function, that has not been previously defined, has been called
19	No RESUME	An error-trapping routine has been entered that contains no RESUME statement
20	RESUME without error	A RESUME statement has been encountered before an error-trapping routine is entered
21	Unprintable error	An error message is not printable i.e. corresponds to an error with an undefined error code
22	Missing operand	An expression contains an operator but no following operand
23	Line buffer overflow	An attempt has been made to enter a line with more than 255 characters
26	FOR without NEXT	A FOR has been encountered without a matching NEXT
29	WHILE without WEND	A WHILE has been encountered without a matching WEND
30	WEND without WHILE	A WEND has been encountered without a matching WHILE
31	IEEE: Invalid talker/ listener address	Use of illegal talker/listener address

32	IEEE: talker = listener address	An attempt has been made to talk to a talker, or listen to a listener
33	IEEE: Unprintable error	An error message is not printable i.e. corresponds to an error with an undefined error code
34	IEEE: Board not present	An attempt has been made to use IEEE on a machine which does not have the optional IEEE interface
25	Window not open	An attempt has been made to use a window which is not at present open
36	Unable to create window	The window to be created is too big or too small for its mode (graphics or text)
37	Invalid action verb	An action verb has been incorrectly spelt or used
38	Parameter out of range	One or more parameters have exceeded the limits set for their range
39	Too many dimensions	An attempt has been made to use an array of more than one dimension, in graphics mode
5Ø	Field overflow	A FIELD statement has attempted to allocate more bytes than were specified for the record length of a random file
51	Internal error (also used in PCOS)	An internal malfunction has occurred. Report the conditions under which the error occurred to your Support Organization
52	Bad file number	A statement or command refers to a file (having a file number not within the range specified at initialisation) or the corresponding file is not open

53	File not found (also used in PCOS)	A LOAD, KILL or OPEN statement refers to a file that does not exist on the current disk
54	Bad file mode (also used in PCOS)	An attempt has been made to use random file operations (GET or PUT) with a sequential file; or to use the sequential operation LOAD with a random file; or to use an illegal file mode with OPEN, i.e. not A,I,O, or R
55	File already open (also used in PCOS)	A sequential OPEN, O has been issued for a file that is already open, or a KILL has been applied to a file that is open
57	Disk I/O error (also used in PCOS)	An input/output error has occurred during a disk I/O operation. It is a termination error, i.e. PCOS/ BASIC cannot recover - apply a RESET
58	File already exists (also used in PCOS)	The file name specified in a NAME statement is identical to a file name already in use on the disk
61	Disk full (also used in PCOS)	All disk storage space available is in use
62	Input past end	An INPUT statement has been executed after all the data has been assigned, or for an empty (null file) Hint: use of EOF function to detect end of file

63	Bad record number	The record number used with a GET or PUT statement exceeds range, i.e. is /or greater than 32767
64	Bad file name	An invalid form of file name has been used with KILL, LOAD, OPEN or SAVE e.g.: - too long - includes illegal characters such as space or hyphen (also used in PCOS)
66	Direct statement in file	A direct (immediate) statement has been encountered when loading an ASCII format file. The LOAD operation is terminated
67	Too many files (also used in PCOS)	An attempt has been made to create a new file (using SAVE or OPEN) when the present directory is already full
68	Internal error	An internal fault has occurred. Report the conditions under which the error occurred to your Support Organization
69	Volume name not found	The volume name referred to does not match (either of) the disk(s) currently inserted
7Ø	Rename error	An attempt has been made to rename a volume with an illegal name
71	Volume number error	The specified volume number is not allowed
72	Volume not enabled	The volume identifier includes a password which must be quoted
73	Invalid password	The password entered is not allowed

74	Illegal disk change	The disk has been changed since last using the file
75	Write protected	An attempt has been made to write to a write protected disk
76	Error in parameter	A parameter contains an illegal character
77	Too many parameters	More than the required number of parameters have been specified
78	File not OPEN	An attempt has been made to access a file that is not open

PCOS ERRORS		
ERROR CODE	DESCRIPTION AS OUTPUT	COMMENT
7	Exceeded memory limit	A program has been called, or a parameter request made, which is too great for the current memory space available
13	Bad data type	A string value has been entered when a numeric value is required, or vice versa
53	File not found	The file referred to cannot be accessed, because it is not on the enabled disk
54	Bad file open mode	
55	File already open	An attempt has been made to open a file that is already open
57	Disk I/O error	An input/output error has occurred during a disk I/O operation. It is a termination error, i.e. PCOS/ BASIC cannot recover - apply RESET

▶

58	File already exists	The file name you are trying to assign is identical to a file name already on disk
6Ø	Disk has not been initialised	The disk accessed has not been initialised
61	Disk is full	All the disk storage space is in use
62	Eof hit unexpectedly	
63	Bad record number	The record number is out of range, i.e. it is either Ø or greater than 32767
64	Bad file name	A file name has been used that is either too long or contains illegal characters
71	Volume name not found	The volume name referred to does not match any disk that is currently inserted
72	Name already exists	The file name specified as the new name when renaming a file already exists since the file was last used
76	Match failed	The specified password does not match that of the file
77	Illegal disk change	The disk has been changed
78	File is write protected	An attempt has been made to write to a file which has write protection
79	Copy protected file	An attempt has been made to access a file that is copy protected
9Ø	Error in parameter	One or more of the quoted parameters contains an unacceptable value

91	Too many parameters	More than the required number of parameters have been specified for a command
92	Command not found	An invalid keyword has been entered
96	File not open	An attempt has been made to access a file that is not open
99	Bad load file	
1Ø1	Bad time or date	An invalid time or date has been entered
1Ø6	Function key already exists	An attempt has been made to PKEY a string to a key which has already been allocated
1Ø8	Call user error	
11Ø	Time out error	
111	Invalid device name	The specified device name does not exist

E. OLISORT STATUS CODES

ABOUT THIS APPENDIX

Here we give you a list of the status codes which may be returned from Olisort.

OLISORT STATUS CODES

When you run Olisort a status code is returned to the calling program via the BASIC variable OLISORT.ERROR%. If you write your own BASIC program you can test this variable in the normal way. If you use the utility olisortx to run Olisort, any status codes which result in abnormal termination of Olisort are automatically reported.

A full list of these status codes is given below.

ERROR CODE	MEANING
1	READ PAST END OF FILE. This indicates a system malfunction or a disk error. Check that your disks are not corrupted or physically damaged.
2	READ UNWRITTEN DATA. As for error code 1.
3	FILE NOT PRESENT WHEN OPENED. A file required by Olisort could not be found on any of the disks loaded. Check your disks to see if they contain all the files you have specified. In the event of disk changes, ensure you are loading the correct disks at the correct time.
4	OUT OF DIRECTORY SPACE. One of the loaded disks has a full directory. Olisort is unable to create one of its files. Check disks to see if you can delete unnecessary files.
5	OUT OF DATA SPACE. Olisort cannot create one of its files because one of the loaded disks is full. Check to see if you can delete any files to release space on the disk, or try and use the Olisort disk change facility.
6	EXTENT ERROR ON WRITE. As for error code 1.
7	FILE NOT PRESENT WHEN CLOSED. As for error code 1.
8	NO MATCH WHEN SEARCHING FOR FILES. As for error code 1.
9	ERROR READING INPUT FILE.
1Ø	DISK READ ERROR in which case there is something physically wrong with one of the disks that Olisort is trying to access, or the meaning is the same as error code 5.
11	Reserved for future use.

12	Reserved for future use.
13	FORMAT ERROR IN THE COMMAND STRING. A Command String contains one, or more, invalid parameters. Check your Command String.
14	SELECT/EXCLUDE KEY LENGTH IS ZERO. A Command String contains an fl parameter for a Select/Exclude Key, which is equal to zero. Check your Command String.
15	As for error code 13.
16	OLISORT PROGRAM NOT PRESENT. The OLISORT disk is not loaded on the drive specified by the sd parameter in the Command String. Check that you have loaded the disks as you meant to, and/or check your Command String.

F. OLISORT PHASES

ABOUT THIS APPENDIX

Here we tell you specific information about the files which Olisort requires at which phase of its processing. This is useful information if you are intending to do disk changes, and important information if you decide to rearrange the files comprising Olisort.

CONTENTS

OLISORT PHASES

There are three phases of processing which Olisort goes through:

- **Input Phase** where the Olisort program is called via a BASIC program or the olisortx utility. Immediately after being called, the BASIC interpreter and the calling program are dumped onto disk in the BDUMP file. Then Olisort begins to read the Input File (or files).

- **Sort Phase** where the sort process takes place. Olisort continues to read the Input File (or files) and creates Sort Work-Files.

- **Output Phase** where Olisort writes the Output File, erases the Sort Work-Files, restores the BASIC interpreter and the calling program, and finally erases the BDUMP file.

OLISORT DISK CHANGES

The Sort Work-File disk change comes after the Input Phase and before the Sort Phase. The Output File disk change comes after the Sort Phase and before the Output Phase.

OLISORT FILES WHICH MUST BE ON-LINE

The files which must be available to Olisort during its particular phases, are as follows:

- During the Input Phase:
  . OLISORT.BAS
  . os.cmd
  . all Input Files
  . the Parameter File (if required)
  . the disk on which the BDUMP file will be written

- For the Sort Phase:
  . all Input Files
  . the disk on which the Sort Work-Files will be written

- For the Output Phase:
  . the disk which contains the Sort Work-Files
  . the disk which contains the BDUMP file
  . the disk on which the Output File will be written

NOTICE

Ing. C. Olivetti & C. S.p.A. reserves the right to make improvements in the product described in this manual at any time and without notice.

This material was prepared for the benefit of Olivetti customers. It is recommended that the package be test run before actual use.

Anything in the standard form of the Olivetti Sales Contract to the contrary not withstanding, all software being licensed to Customer is licensed "as is". THERE ARE NO WARRANTIES EXPRESS OR IMPLIED INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTY OF FITNESS FOR PURPOSE AND OLIVETTI SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES IN CONNECTION WITH SUCH SOFTWARE.

GU Code 3982260 E (0)

Printed in Italy

GU Code 3982260 E (0)

Printed in Italy