

Wsort

User's Reference Manual

188-190-204 A

May 1985

WICATsystems



Copyright © 1985 by WICAT Systems Incorporated
All Rights Reserved
Printed in the United States of America

Receipt of this manual must not be construed as any kind of commitment,
on the part of WICAT Systems Incorporated, regarding delivery or
ownership of items manufactured by WICAT.

This manual is subject to change without notice.

first printing May 1985

Typographical Conventions Used in this Publication

Bold facing indicates what you should type.

Square brackets, [], indicate a function key, the name of which appears in uppercase within the brackets. For example, [RETRN], [CTRL], etc.

Underlining is used for emphasis.

Information about this Manual

Review the following items before you read this publication.

The subject of this manual

This manual explains what the WSORT program is and how to execute WSORT.

The audience for whom this publication was written

This manual is for the user or system manager who needs to sort the contents of text files. Before you read this manual, you should complete the tutorials in either the WMCS User's Introductory Manual or the WMCS System Manager's Introductory Manual.

Related publications

The chart on the following page lists other publications about the WMCS and the order in which they should be read.

Reader's Guide to WMCS Publications

Instructions: Determine the audience to which you belong and then read *only* the publications at an *arrowhead*.

Dotted arrowheads indicate optional reading.

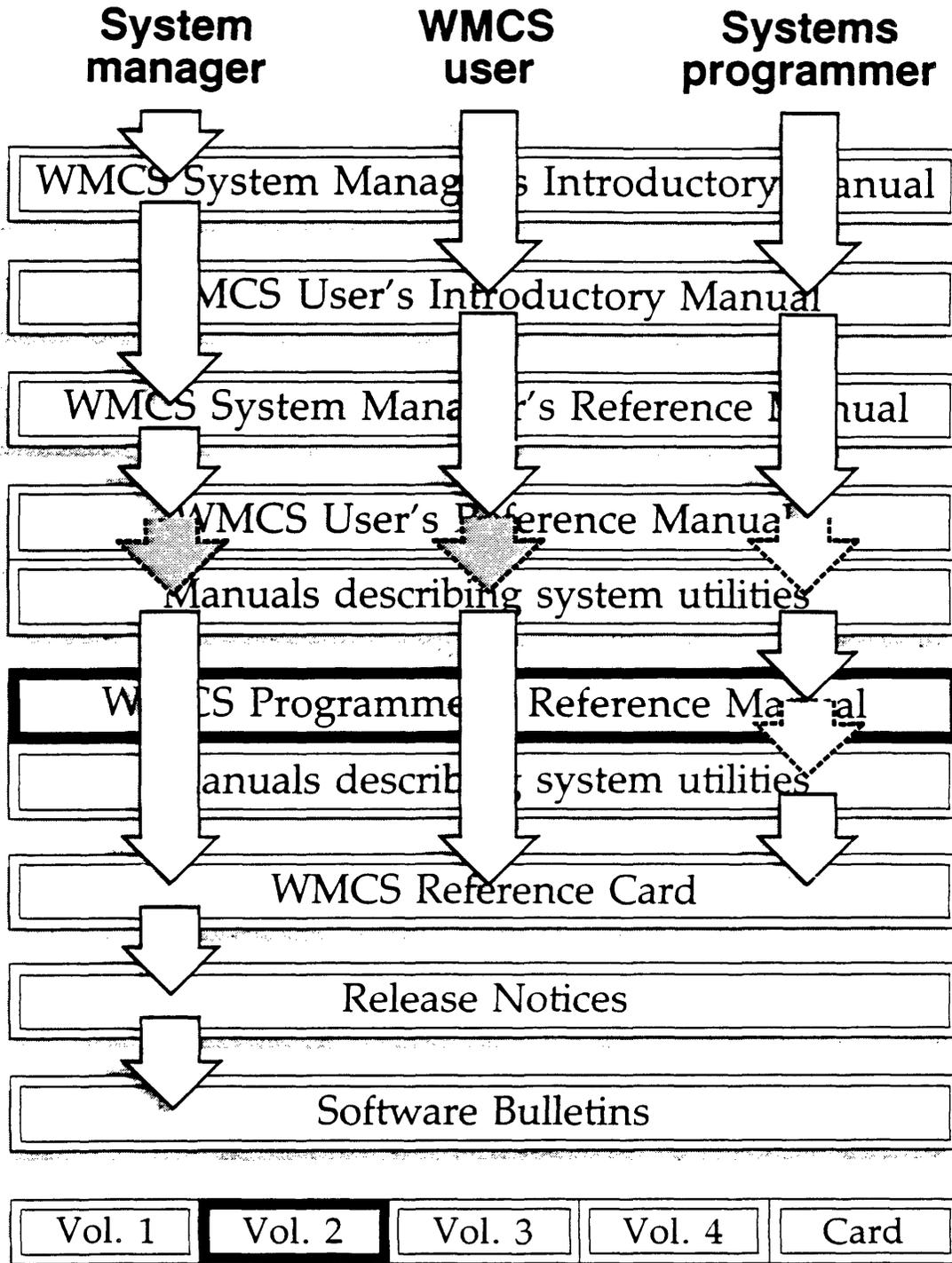


Table of Contents

Chapter 1 General Information

How WSORT works.....	1-1
How to Execute the WSORT Program.....	1-2
Parameters.....	1-2
Switches:.....	1-2
Examples.....	1-6

Chapter 2 WSORT Key Modifiers

Direction Modifiers.....	2-1
Field Modifiers.....	2-2
Field Submodifiers.....	2-7
Data Type Modifiers.....	2-11
Miscellaneous Modifiers.....	2-15

Chapter 1

General Information

The WSORT program, one of the utilities in the WMCS operating system, sorts the contents of one or more files according to specified keys. WSORT can also merge the sorted files.

This program is similar to the SORT program described in the WMCS User's Reference Manual. Unlike SORT, the switches and key modifiers in WSORT are written in the standard WMCS syntax. WSORT is faster and generally much easier to use than SORT.

How WSORT works

WSORT rearranges the contents of a file or files according to the criteria you specify. The sorted result is put into a destination file. A sort "key" is used to specify what information in the record should be compared with other records to sort that record. In a personnel list, for example, you can specify that the first key is the department number and that the second key is the salary. The sorted result is a list of employees ordered by salary within each department. If you do not specify a sort key, the entire record is used as the key.

Records can be divided into fields. Fields give you the ability to describe how to find a key, when the exact position of the key in the record may vary. For example, the parts of an address may be separated on a line by commas. You can specify that a field is delimited by commas. This will enable you to sort the addresses by city name, which is somewhere after the second comma. The exact position of the sort key (city name) can be described for each record (address) by describing the field (after the second comma) it is in.

Sort keys are specified with the :keyN= switch. The "N" is a number from 1 to 10 that determines the order the keys are used. "Modifiers" follow the equal sign in the switch. A key modifier describes the limits of a key. Submodifiers further pinpoint the location of the key. Key modifiers

General Information

for the :keyN= switch are described in detail in chapter 2 of this manual.

How to Execute the WSORT Program

A summary of how to execute the WSORT command, useful for quick reference, is provided in the WMCS User's Reference Manual. A complete description of executing the WSORT program is given here.

Parameters

There are two required parameters, File List and Destination. They appear on the command line in the following order:

```
> wsort filelist destination :switches
```

The File List parameter specifies the names of the source files whose contents are to be sorted. It can contain several filenames separated by commas. Wildcard symbols are allowed in the File List parameter.

The Destination parameter is a single file designation into which the sorted contents of the source files are merged and written. No wildcard symbols are allowed in this parameter.

Switches

The following switches apply to the entire sort process and control how the files are sorted and how the records are treated. All switches allow unique abbreviations. Switches may appear in any order on the command line.

These five switches are standard switches which affect the selection of files when wildcard symbols are used in the File List parameter:

:before=	Files created or modified before the date and time you specify in this switch are included in the File List.
:since=	Files created or modified since the date and time you specify in this switch are included in the File List.
:exclude=	Files in the list you specify in this switch are excluded from the File List parameter.

General Information

- :uic=** Files owned by the UICs or usernames you specify in this switch are included in the File List.
- :mod** The modification date is used instead of the creation date in the **:before=** and **:since=** switches.

These switches control how files are sorted and how results of the sort are displayed.

- :sort=** Files that match the wildcard symbols in the File List are read in the order you specify in this switch. Type **extension**, **uic**, or **date**. The default is alphabetical order by filename.
- :log** Names of the source files are logged to SYS\$ERROR unless you specify **:nolog**.
- :duplicates** Records with duplicate key values are output unless you specify **:noduplicates**. If **:noduplicates** is specified, only the first record of several records with duplicate sort key values is written to the Destination file.
- :stats** Use this switch to display statistics about the sort to SYS\$OUTPUT. The display includes the number of files sorted, the number of records sorted, the number of comparisons performed, the amount of memory used (in Kbytes), and the amount of time used (in seconds) during the sort.
- :errorhandling=** This switch controls how erroneous records are handled. Six mutually exclusive options are available. No abbreviations are allowed.
- The default value is **:error=reject**. The values "report" and "stop" force the **:log** switch to **:log** even if **:nolog** has been explicitly specified.
- Type one of the following values for this switch:
- reject** Do not report erroneous records and exclude them from the sort.
- stop** Report the first erroneous record and then terminate.

General Information

report Report all erroneous records. If the `:errorfile=` switch is used, write the sorted erroneous records to that file.

ignore Ignore erroneous records and do not report them. Sort them with the other records. If the `:errorfile=` switch is used, write the sorted erroneous records to that file.

first Sort erroneous records as a group and write them at the top of the Destination file. If the `:errorfile=` switch is used, write the sorted erroneous records to that file.

last Sort erroneous records as a group and write them at the bottom of the Destination file. If the `:errorfile=` switch is used, write the sorted erroneous records to that file.

`:errorfile=` Use this switch to specify the name of a file in which the sorted erroneous records are placed. If the value of the `:errorhandling=` switch is "first," "last," "ignore" or "report," the sorted erroneous records are not written to the Destination file but are written to this "error" file.

`:text` Source files are treated as text files unless you specify `:notext`. A text file is a file of ASCII characters organized as lines of text. The ASCII `<LF>` character is the line terminator. A non-text file does not have the ASCII `<LF>` character as a line terminator and is organized solely as records with a fixed record length.

`:recordlen=` Use this switch to specify the record size of non-text source and destination files, regardless of the actual file record sizes. Type a value from 1 to 65535 inclusive.

If `:recordlen=` is specified and `:text` is not explicitly specified, or `:notext` is specified, the files are read with the record length you specify in this switch.

If `:notext` is specified and `:recordlen=` is not specified, the record length used for all source

General Information

files is the actual record length of the first source file.

`:maxrecordlen=` Use this switch to specify the maximum length (in bytes) of any record in the source file list. If a text record is read which is longer than the value of this switch, an error is reported and WSORT terminates. The default is `:maxrecordlen=1024`.

`:memory=` Use this switch to specify the maximum amount of memory (in Kbytes) to be used on the initial buffer allocation for sorting records.

If files with fixed record lengths are sorted, the memory used will be 1 Kbyte more than the value specified by this switch.

If text files with different length lines are sorted, the memory used will vary because of the nature of dynamic memory. However, the initial memory allocation will not be more than that specified in this switch.

The more memory that is allocated to the WSORT utility for sorting records, the faster the sort will be. The default is `:memory=64`.

`:ascending` Files are sorted in ascending order unless you specify `:descending`.

`:descending` Use this switch to sort files in descending order. This switch is mutually exclusive to the `:ascending` switch.

`:delimiter=` Use this switch to specify the characters which delimit fields within a record. This switch is only valid when used with the `field=` key modifier. (See chapter 2.) The value of the `field=` key modifier selects the field from which the sort key is extracted. Type a string of characters enclosed in double quotation marks. There are no default characters for this switch.

`:skip=` Use this switch to specify the characters which are skipped when scanning a record for a key. Type a string of characters enclosed in double quotation marks. The default is `:skip=<SP><TAB><LF>`.

General Information

`:ignoreleading=` Use this switch to specify the leading characters which will be ignored when WSORT scans for a key within a field. Type a string of characters enclosed in double quotation marks. The default is `:ignoreleading=<SP><TAB><LF>`.

`:keyN=` Use this switch to specify sort keys upon which the sort is performed. The sort key order is specified by the value of N in the switch name, i.e., the primary sort key switch is `:key1=`, the secondary sort key switch is `:key2=`, etc. Up to 10 keys may be specified. This switch contains modifiers which describe the sort key characteristics. The key modifiers are described in detail in chapter 2 of this manual.

All `:keyN=` switches must be numbered consecutively. For example, if you are using three sort keys, you must specify `:key1=`, `:key2=`, and `key3=`. You cannot specify `:key1=`, `:key2=`, and `:key4=`.

The order in which the `:keyN=` switches appear on the command line does not matter. If there are no key switches specified, the entire record is used as the key and the comparisons are done as ASCII characters.

Examples

The following examples show how various WSORT switches can be used to sort text files.

```
> wsort file.txt sorted.txt :descending
```

Suppose this is the contents of FILE.TXT:

General Information

Directory listing of _DS0/SYSHLP/

File name	Logical/	Physical	Created	Username
ALLOC.HLP.1	2.5/	3.0	16-Apr-1985 00:00:00	SYSTEM
ARCH.HLP.1	5.8/	6.0	12-Oct-1985 00:00:00	SYSTEM
ASCII.HLP.1	3.0/	3.0	12-Oct-1985 00:00:00	SYSTEM
ASSIGN.HLP.1	1.7/	2.0	12-Oct-1985 00:00:00	SYSTEM
BACKUP.HLP.1	3.2/	4.0	12-Oct-1985 00:00:00	SYSTEM
BKUP.HLP.1	1.3/	2.0	12-Oct-1985 00:00:00	SYSTEM
BTUP.HLP.1	3.0/	3.0	12-Oct-1985 00:00:00	SYSTEM
CD.HLP.1	1.4/	2.0	12-Oct-1985 00:00:00	SYSTEM

WSORT sorts FILE.TXT in descending order and puts the result in the file SORTED.TXT. This is the contents of SORTED.TXT:

File name	Logical/	Physical	Created	Username
Directory listing of _DS0/SYSHLP/				
CD.HLP.1	1.4/	2.0	12-Oct-1985 00:00:00	SYSTEM
BTUP.HLP.1	3.0/	3.0	12-Oct-1985 00:00:00	SYSTEM
BKUP.HLP.1	1.3/	2.0	12-Oct-1985 00:00:00	SYSTEM
BACKUP.HLP.1	3.2/	4.0	12-Oct-1985 00:00:00	SYSTEM
ASSIGN.HLP.1	1.7/	2.0	12-Oct-1985 00:00:00	SYSTEM
ASCII.HLP.1	3.0/	3.0	12-Oct-1985 00:00:00	SYSTEM
ARCH.HLP.1	5.8/	6.0	12-Oct-1985 00:00:00	SYSTEM
ALLOC.HLP.1	2.5/	3.0	16-Apr-1985 00:00:00	SYSTEM

> wsort file.txt sorted.txt :stats

Suppose this is the contents of FILE.TXT:

DEVICES.HLP.1	3.6/	4.0	12-Oct-1985 00:00:00	SYSTEM
CONFIG.HLP.2	1.7/	2.0	13-Oct-1985 08:03:12	SYSTEM
COUNT.HLP.1	1.8/	2.0	16-Apr-1985 00:00:00	SYSTEM
DEF.HLP.1	1.1/	2.0	12-Oct-1985 00:00:00	SYSTEM
CRD.HLP.1	1.8/	2.0	16-Apr-1985 00:00:00	SYSTEM
CREATE.HLP.1	1.5/	2.0	12-Oct-1985 00:00:00	SYSTEM
DATES.HLP.1	4.5/	5.0	12-Oct-1985 00:00:00	SYSTEM
DEL.HLP.1	2.0/	2.0	16-Apr-1985 00:00:00	SYSTEM
DEALLOC.HLP.1	1.8/	2.0	16-Apr-1985 00:00:00	SYSTEM
CONFIG.HLP.1	1.7/	2.0	12-Oct-1985 00:00:00	SYSTEM
DEL.HLP.2	2.0/	2.0	16-Apr-1985 08:07:35	SYSTEM
COPY.HLP.1	3.4/	4.0	12-Oct-1985 00:00:00	SYSTEM
DEV.HLP.1	1.7/	2.0	16-Apr-1985 00:00:00	SYSTEM
DEL.HLP.3	2.0/	2.0	16-Apr-1985 09:14:27	SYSTEM

WSORT sorts FILE.TXT in ascending order and puts the result in the file SORTED.TXT. This is the contents of SORTED.TXT:

General Information

CONFIG.HLP.1	1.7/	2.0	12-Oct-1985	00:00:00	SYSTEM
CONFIG.HLP.2	1.7/	2.0	13-Oct-1985	08:03:12	SYSTEM
COPY.HLP.1	3.4/	4.0	12-Oct-1985	00:00:00	SYSTEM
COUNT.HLP.1	1.8/	2.0	16-Apr-1985	00:00:00	SYSTEM
CRD.HLP.1	1.8/	2.0	16-Apr-1985	00:00:00	SYSTEM
CREATE.HLP.1	1.5/	2.0	12-Oct-1985	00:00:00	SYSTEM
DATES.HLP.1	4.5/	5.0	12-Oct-1985	00:00:00	SYSTEM
DEALLOC.HLP.1	1.8/	2.0	16-Apr-1985	00:00:00	SYSTEM
DEF.HLP.1	1.1/	2.0	12-Oct-1985	00:00:00	SYSTEM
DEL.HLP.1	2.0/	2.0	16-Apr-1985	00:00:00	SYSTEM
DEL.HLP.2	2.0/	2.0	16-Apr-1985	08:07:35	SYSTEM
DEL.HLP.3	2.0/	2.0	16-Apr-1985	09:14:27	SYSTEM
DEV.HLP.1	1.7/	2.0	16-Apr-1985	00:00:00	SYSTEM
DEVICES.HLP.1	3.6/	4.0	12-Oct-1985	00:00:00	SYSTEM

The :stats switch displays the following report during the sort:

```
1 file sorted
14 records sorted
52 comparisons performed
22 Kbytes of memory used
1 second of time used
```

> wsort file1.txt,file2.txt,file3.txt sorted.txt :noduplicates

Suppose this is the contents of FILE1.TXT:

Directory listing of _DS0/SYSHLP/

File name	Logical/	Physical	Created	Username
ASCII.HLP.1	3.0/	3.0	12-Oct-1985 00:00:00	SYSTEM
ASSIGN.HLP.1	1.7/	2.0	12-Oct-1985 00:00:00	SYSTEM
BACKUP.HLP.1	3.2/	4.0	12-Oct-1985 00:00:00	SYSTEM
ARCH.HLP.1	5.8/	6.0	12-Oct-1985 00:00:00	SYSTEM
BKUP.HLP.1	1.3/	2.0	12-Oct-1985 00:00:00	SYSTEM
ALLOC.HLP.1	2.5/	3.0	16-Apr-1985 00:00:00	SYSTEM
CD.HLP.1	1.4/	2.0	12-Oct-1985 00:00:00	SYSTEM
BTUP.HLP.1	3.0/	3.0	12-Oct-1985 00:00:00	SYSTEM

Suppose this is the contents of FILE2.TXT:

General Information

CD.HLP.1	1.4/	2.0	12-Oct-1985	00:00:00	SYSTEM
BTUP.HLP.1	3.0/	3.0	12-Oct-1985	00:00:00	SYSTEM
BKUP.HLP.1	1.3/	2.0	12-Oct-1985	00:00:00	SYSTEM
BACKUP.HLP.1	3.2/	4.0	12-Oct-1985	00:00:00	SYSTEM
ASSIGN.HLP.1	1.7/	2.0	12-Oct-1985	00:00:00	SYSTEM
ASCII.HLP.1	3.0/	3.0	12-Oct-1985	00:00:00	SYSTEM
ARCH.HLP.1	5.8/	6.0	12-Oct-1985	00:00:00	SYSTEM
ALLOC.HLP.1	2.5/	3.0	16-Apr-1985	00:00:00	SYSTEM

Suppose this is the contents of FILE3.TXT:

FUNCTIONS.HLP.1	1.8/	2.0	12-Oct-1985	00:00:00	SYSTEM
DM.HLP.1	1.5/	2.0	12-Oct-1985	00:00:00	SYSTEM
EDITLIST.HLP.1	1.8/	2.0	12-Oct-1985	00:00:00	SYSTEM
DMNT.HLP.1	1.9/	2.0	12-Oct-1985	00:00:00	SYSTEM
DUMP.HLP.1	2.5/	3.0	12-Oct-1985	00:00:00	SYSTEM
ERR.HLP.1	1.2/	2.0	12-Oct-1985	00:00:00	SYSTEM
FILELIST.HLP.1	1.8/	2.0	12-Oct-1985	00:00:00	SYSTEM
FSTAT.HLP.1	3.2/	4.0	16-Apr-1985	00:00:00	SYSTEM
FPMGR.HLP.3	1.8/	2.0	04-Mar-1985	00:00:00	SYSTEM
DSTAT.HLP.1	8.3/	9.0	12-Oct-1985	00:00:00	SYSTEM
FIXSTACK.HLP.1	1.5/	2.0	16-Apr-1985	00:00:00	SYSTEM

All three files are sorted and the results are merged and put into the file SORTED.TXT. The :noduplicates switch suppresses the display of duplicate records in the sort. This is the contents of SORTED.TXT:

General Information

ALLOC.HLP.1	2.5/	3.0	16-Apr-1985	00:00:00	SYSTEM
ARCH.HLP.1	5.8/	6.0	12-Oct-1985	00:00:00	SYSTEM
ASCII.HLP.1	3.0/	3.0	12-Oct-1985	00:00:00	SYSTEM
ASSIGN.HLP.1	1.7/	2.0	12-Oct-1985	00:00:00	SYSTEM
BACKUP.HLP.1	3.2/	4.0	12-Oct-1985	00:00:00	SYSTEM
BKUP.HLP.1	1.3/	2.0	12-Oct-1985	00:00:00	SYSTEM
BTUP.HLP.1	3.0/	3.0	12-Oct-1985	00:00:00	SYSTEM
CD.HLP.1	1.4/	2.0	12-Oct-1985	00:00:00	SYSTEM
CHECKSUM.HLP.1	1.9/	2.0	16-Apr-1985	00:00:00	SYSTEM
CHKD.HLP.1	1.8/	2.0	12-Oct-1985	00:00:00	SYSTEM
CIP.HLP.1	1.4/	2.0	12-Oct-1985	00:00:00	SYSTEM
CMD.HLP.1	1.8/	2.0	12-Oct-1985	00:00:00	SYSTEM
DM.HLP.1	1.5/	2.0	12-Oct-1985	00:00:00	SYSTEM
DMNT.HLP.1	1.9/	2.0	12-Oct-1985	00:00:00	SYSTEM
DSTAT.HLP.1	8.3/	9.0	12-Oct-1985	00:00:00	SYSTEM
DUMP.HLP.1	2.5/	3.0	12-Oct-1985	00:00:00	SYSTEM
Directory listing of _DS0/SYSHLP/					
EDITLIST.HLP.1	1.8/	2.0	12-Oct-1985	00:00:00	SYSTEM
ERR.HLP.1	1.2/	2.0	12-Oct-1985	00:00:00	SYSTEM
FILELIST.HLP.1	1.8/	2.0	12-Oct-1985	00:00:00	SYSTEM
FIXSTACK.HLP.1	1.5/	2.0	16-Apr-1985	00:00:00	SYSTEM
FPMGR.HLP.3	1.8/	2.0	04-Mar-1985	00:00:00	SYSTEM
FSTAT.HLP.1	3.2/	4.0	16-Apr-1985	00:00:00	SYSTEM
FUNCTIONS.HLP.1	1.8/	2.0	12-Oct-1985	00:00:00	SYSTEM
File name	Logical/	Physical	Created		Username

For examples of the :keyN= switch, read chapter 2 of this manual.

Chapter 2

WSORT Key Modifiers

This chapter explains the key modifiers that can be used with the `:keyN=` switch for WSORT. Key modifiers allow you to perform complicated sorts with custom-made keys. If you do not use the `:keyN=` switch, the entire record is used as the key and the comparisons are done as ASCII characters.

As many as ten `:keyN=` switches can be typed on the command line. (N is a number from 1 to 10.) The value of N determines the sort key order. For example, `:key1=` is the primary sort key, `:key2=` is the secondary sort key, etc. All `:keyN=` switches must be numbered consecutively. However, they can be typed in any order on the command line.

Key modifiers are typed after the equal sign and are separated by commas. They can appear in any order. For example:

```
> wsort file.txt sorted.txt :key1=delimiter=" ",field=1,dictionary
```

Unique abbreviations of the modifiers are allowed. For example, "asc" for ascending, "dic" for dictionary, "f" for floatingpoint, etc.

Key modifiers override WSORT switches and default values with similar names. For example, `:key1=ascending` overrides the `:descending` switch on the command line.

Direction Modifiers

The following mutually exclusive modifiers specify the direction of the sort for a single sort key.

WSORT Key Modifiers

ascending	Use this modifier to specify that the key will sort in ascending order.
descending	Use this modifier to specify that the key will sort in descending order.

Field Modifiers

Sort keys can be used to divide a record into fields. Fields can be defined in a very flexible manner using key modifiers. A sort key is extracted from a field based upon the modifiers and data type of the key.

Field definition can be as simple as specifying an absolute location in a record, or as complex as isolating a free-format, anywhere-in-the-record field based upon the data in a record.

The following sample file record will be used to illustrate the operation of field modifiers. Results of each field definition example will be enclosed in double quotation marks in order to make spaces easier to see. The sample record is a single line from the display of this DIR command:

```
> dir * :size :create :username :head
```

The sample record is printed below in boldface. The numbers above the sample line represent the relative byte positions.

```
11111111112222222222333333333344444444445555555555666666  
1234567890123456789012345678901234567890123456789012345  
  
SORT.C.242                    85.4/            86.0 23-Oct-1985 10:24:57 GB
```

The following modifiers govern the extraction of a single sort key from a record:

startat=	Use this modifier to specify (by byte position) where the field scan will start, relative to the beginning of the record. Any data before that specified by this modifier will not be examined when the fields are defined. This modifier can have positional submodifiers. The default is "startat=1" (the first byte in the record).
----------	--

Suppose you typed this key modifier:

```
:keyl=startat=36
```

This defines the following part of the sample record as the key field:

```
" 86.0 23-Oct-1985 10:24:57 GB"
```

endat=n

Use this modifier to specify (by byte position) where the field scan will end, relative to the beginning of the record. Any data after that specified by this modifier will not be examined when the fields are defined. This modifier can have positional submodifiers. The default is the end of the record.

Suppose you typed this key modifier:

```
:keyl=endat=15
```

This defines the following part of the sample record as the key field:

```
"SORT.C.242      "
```

Suppose you typed this key modifier:

```
:keyl=startat=24,endat=30
```

This defines the following part of the sample record as the key field:

```
" 85.4"
```

skip=

Use this modifier to specify the characters to skip when parsing a key from a text record. Type a string of characters enclosed in double quotation marks. When a "field=" modifier is specified for a sort key, the characters specified by this modifier are skipped before the search for a character specified by the :delimiter= switch. The default is the value of the :skip= switch.

WSORT Key Modifiers

Suppose you typed this key modifier:

```
:keyl=startat=24,endat=30,skip=" "
```

This defines the following part of the sample record as the key field:

```
"85.4"
```

`delimiter=`

Use this modifier to specify the characters which delimit fields in a text record. The value of this modifier is a string of characters enclosed in double quotation marks. When a "field=" modifier is specified for a sort key, the characters specified by this modifier are used to divide the record into fields. Since delimiters can be in different places relative to the beginning of a record for different records, fields can vary in size and number. The default is the value of the ":delimiter=" switch.

Suppose you typed this key modifier:

```
:keyl=delimiter="./"
```

This divides the sample record into eight fields. Note that the delimiter characters are not part of the fields. (The sample fields are enclosed in double quotation marks.)

```
field 1 = "SORT"  
field 2 = "C"  
field 3 = "242"           85"  
field 4 = "4"  
field 5 = "      86"  
field 6 = "0 23-Oct-1985 10"  
field 7 = "24"  
field 8 = "57 GB"
```

Because delimiter characters are not contained in fields, it is possible to have a field with a length of zero bytes created by two adjacent delimiters. This is known as an "empty" field.

WSORT Key Modifiers

Suppose you typed this key modifier:

```
:keyl=startat=54,endat=63,delimiter=" :"
```

This divides the sample record into five fields:

```
field 1 = ""          (empty field)
field 2 = "10"
field 3 = "24"
field 4 = "57"
field 5 = ""          (empty field)
```

Any character specified by the "delimiter=" modifier can not appear in the "ignoreleading" modifier for the same sort key definition.

field=

Use this modifier to specify which field the key resides in. The first field in a record is 1, the second field is 2, etc. There is no default value for this modifier. Therefore, a record is not divided into fields unless this modifier is explicitly specified, even if the "delimiter=" modifier is specified.

Suppose you typed this key modifier:

```
:keyl=startat=54,endat=63,\
delimiter=" :",field=2
```

This defines the following part of the sample record as the key field:

```
"10"
```

Since the number of fields in a record can vary, any record which divides into fewer fields than is specified by this modifier results in an erroneous record. Selection of an empty field by this modifier also results in an erroneous record. The disposition of an erroneous record is specified by the :errorhandling= switch.

length=

Use this modifier to place an absolute limit (in bytes) on the field length. There is no default length. If no length is specified, it is determined by the other modifiers. This modifier can have length submodifiers.

WSORT Key Modifiers

Suppose you typed this key modifier:

```
:keyl=startat=24,endat=30,length=5
```

This defines the following part of the sample record as the key field:

```
" 85"
```

Suppose you typed this key modifier:

```
:keyl=length=19
```

This defines the following part of the sample record as the key field:

```
"SORT.C.242      "
```

offset=

Use this modifier to specify where the key starts, relative to the beginning of a field. This modifier can use positional submodifiers. The default is "offset=0".

Suppose you typed this key modifier:

```
:keyl=length=25,offset=5
```

This defines the following part of the sample record as the key field:

```
"C.242          85.4"
```

Suppose you typed this key modifier:

```
:keyl=delimiter="/",field=2,offset=-7,\  
length=6
```

This defines the following part of the sample record as the key field:

```
" 85.4"
```

ignoreleading=

Use this modifier to specify the leading characters which are ignored when extracting the key from a field. Type a string of characters enclosed in double quotation marks. Characters are ignored starting from the offset from the beginning of the field until the first non-

ignorable character is encountered. The default value is the value specified by the :ignoreleading= switch.

Suppose you typed this key modifier:

```
:keyl=startat=24,endat=32,ignoreleading=" "
```

This defines the following part of the sample record as the key field:

```
"85.4/ "
```

Suppose you typed this key modifier:

```
:keyl=delim="/",field=2,off=-7,len=6,ignoreleading=" "
```

This defines the following part of the sample record as the key field:

```
"85.4"
```

Any character in the "ignoreleading=" modifier cannot appear in the "delimiter=" modifier for the same sort key definition.

Field Submodifiers

Positional submodifiers are used for specifying the value of the "startat=", "endat=", "length=" and "offset=" key modifiers. If more than one positional submodifier is used, they must be separated by commas and enclosed in double quotation marks. For example:

```
:keyl=startat="byte=2,bit=4"
```

Positional submodifiers specify a relative position, i.e., 1 means the first, 2 means the second, etc. There are eight different submodifiers.

(number) This submodifier is a number which implies the relative byte position. The following positional submodifier specifies that the field starts at the 17th byte of a record:

```
startat=17
```

WSORT Key Modifiers

- `bcddigits=` This submodifier is a synonym for "nibbles=".
- `bits=` This submodifier specifies the relative bit position. The following positional submodifier specifies that the key offset starts at the 5th bit from the beginning of the field:
- `offset=bits=5`
- The following examples are all equivalent:
- `startat=bits=17`
`startat=bytes=3`
`startat=3`
- `bytes=` This submodifier specifies the relative byte position. A byte is 8 bits. The following positional submodifier specifies that the field ends in the 32nd byte of a record:
- `endat=bytes=32`
- `characters=` This submodifier is a synonym for "bytes=".
- `doublewords=` This submodifier specifies the relative doubleword position. A doubleword is 32 bits. There are four bytes per doubleword. The following positional submodifier specifies that the key offset starts at the 2nd doubleword from the beginning of the field:
- `offset=doubleword=2`
- The following submodifier specifies the same relative position:
- `offset=bytes=5`
- `nibbles=` This submodifier specifies the relative nibble position. A nibble is 4 bits. There are two nibbles per byte. The following submodifier specifies that the field starts at the 3rd nibble of a record:

startat=nibbles=3

The following submodifier specifies the same relative position:

startat=bytes=2

words=

This submodifier specifies the relative word position. A word is 16 bits. There are two bytes per word. The following submodifier specifies that the field ends at the 6th word of a record:

endat=words=6

The following example specifies the same relative position:

startat=bytes=11

If several positional submodifiers are used together, they are relative to each other, with the first being relative to the beginning of the record. The following sets of equivalent examples illustrate the relativity. The final position is marked with an asterisk in the "record" line.

```
startat="doublewords=1,words=1,bytes=1,nibbles=1,bits=1"
startat="word=1,bit=1"
startat=byte=1
startat=1
```

Record	*																		
Doublewords	1			2				3											
Words	1		2		3		4		5		6								
Bytes	1	2	3	4	5	6	7	8	9	10	11	12							

```
startat="doublewords=2,words=2,bytes=2,nibbles=2"
startat="word=4,bit=13"
startat=nibbles=16
```

Record																			*
Doublewords	1			2				3											
Words	1		2		3		4		5		6								
Bytes	1	2	3	4	5	6	7	8	9	10	11	12							
Nibbles																			

Data Type Modifiers

The following key modifiers specify the data type of a single sort key. These can further refine the value of the sort key after it has been extracted from a field. There are thirteen data types used by the WSORT utility. Each is mutually exclusive with the others.

The following sample file record will be used to illustrate the operation of data type modifiers. The sample record is a single line from the display of this DIR command:

```
> dir * :size :create :username :head
```

The sample record is printed below in boldface. The numbers above the sample line represent the relative byte positions.

```

11111111112222222222333333333344444444445555555555666666
1234567890123456789012345678901234567890123456789012345
SORT.C.242                85.4/          86.0 23-Oct-1985 10:24:57 GB

```

The following modifiers govern the extraction of a single sort key from a record:

ascii Use this modifier to specify the sort key data type as ASCII (8-bit unsigned byte) characters. This is the default data type when a data type modifier is not specified. This data type does not further modify the sort key as defined by the field definition modifiers.

asciidecimal Use this modifier to specify the sort key data type as an ASCII decimal number. A plus, +, or a minus, -, sign can appear immediately before the first numeric character. The key value is converted to a signed 32-bit binary integer for comparison. Any leading spaces or tabs are skipped when scanning the field for the ASCII decimal number. This is in addition to those specified by the "ignoreleading=" modifier.

WSORT Key Modifiers

Suppose you typed this key modifier:

```
:key1=del="/",field=2,off=-7,len=6,\
  asciidecimal
```

This defines the following integer value from the sample record as the sort key:

85

asciifloatingpoint Use this modifier to specify the sort key data type as an ASCII floating-point number. A plus or a minus sign can appear immediately before the first numeric character. The number can be in fixed point (x.xx) or scientific (x.xxExx) notation. The key value is converted to a signed 64-bit binary floating-point number for comparisons. Any leading spaces or tabs are skipped when scanning the field for the ASCII floating-point number. This is in addition to those specified by the "ignoreleading=" modifier.

Suppose you typed this key modifier:

```
:key1=del="/",field=2,off=-7,len=6,asciifloa
tingpoint
```

This defines the following floating-point value from the sample record as the sort key:

85.4

asciihexadecimal This modifier specifies the sort key data type as an ASCII hexadecimal number. A dollar sign, \$, can appear immediately before the first hexadecimal character. Uppercase or lowercase characters can be used as hexadecimal digits. The key value is converted to a signed 32-bit binary integer for comparison. Any leading spaces or tabs are skipped when scanning the field for the ASCII hexadecimal number. This is in addition to those specified by the "ignoreleading=" modifier.

WSORT Key Modifiers

Suppose you typed this key modifier:

```
:key1=del="/",field=2,off=-7,len=6,asciihexa  
decimal
```

This defines the following integer value (base 10) from the sample record as the sort key:

```
133 (85 in base 16)
```

- asciireal** This is a synonym for "asciifloatingpoint".
- binary** Use this modifier to specify the sort key data type as a signed binary integer. The default length of the binary data type is four bytes.
- bitfield** Use this modifier to specify the sort key data type as an unsigned bit field. It can be from 1 to 32 bits in length and does not need to begin or end on byte boundaries. The default length of a bit field is 32 bits.
- caseless** This modifier is the same as "ascii" except that no distinction is made between uppercase and lowercase alphabetic characters during comparisons. If two records have the same key value, their order is determined by comparing the original fields as ASCII characters.
- dictionary** Use this modifier to specify the sort key data type as alphanumeric. All alphanumeric characters in the field form the value of the key. If two records have the same key value, their order is determined by comparing the original fields as ASCII characters.

WSORT Key Modifiers

Suppose you typed this key modifier:

```
:key1=delimiter=" ",field=1,dictionary
```

This defines the following part of the sample record as the value of the sort key:

```
"sortc242"
```

floatingpoint

Use this modifier to specify the sort key data type as a signed binary floating-point number. The length of the floating-point number can be either four bytes (32 bits) or eight bytes (64 bits). The floating point format conforms to the IEEE floating-point standard.

location

Use this modifier to specify the sort key data type is actually the location of the key in the record. The location is relative to the beginning of the record.

Suppose you typed this key modifier:

```
:key1=del="/",field=2,off=-7,len=6,\  
ign=" ",location
```

This defines the following binary integer value from the sample record as the sort key:

```
27
```

real

This modifier is a synonym for "floatingpoint".

width

Use this modifier to specify the sort key data type is actually the width, or length, of the field in the record.

Suppose you typed this key modifier:

```
:key1=del="/",field=2,off=-7,len=6,  
    ign=" ",width
```

This defines the following binary integer value from the sample record as the sort key:

4

Miscellaneous Modifiers

The following modifiers perform miscellaneous functions for a single sort key.

allowerrors Use this modifier to specify whether or not a record is marked erroneous if an error occurs while extracting the sort key. The default is "noallowerrors", which marks a record as erroneous if an error occurs during sort key extraction.

ignoreleading Use this modifier to specify whether or not leading characters are ignored when extracting a sort key from a record field. This modifier can override the value specified by the :ignoreleading= switch and cause leading characters to not be ignored when extracting a key from a field.

Note that this modifier is different from the "ignoreleading=" valued modifier. The default value of this modifier is "ignoreleading". This allows the file default to be applied to the sort key if no "ignoreleading=" modifier has been specified.

WICAT Systems, Inc.

Product-documentation Comment Form

We are constantly improving our documentation, and we welcome specific comments on this manual.

Document Title: _____

Part Number: _____

- Your Position:**
- | | |
|--|--|
| <input type="checkbox"/> Novice user | <input type="checkbox"/> System manager |
| <input type="checkbox"/> Experienced user | <input type="checkbox"/> Systems analyst |
| <input type="checkbox"/> Applications programmer | <input type="checkbox"/> Hardware technician |

Questions and Comments

Page No.

Briefly describe examples, illustrations, or information that you think should be added to this manual.

_____	_____
_____	_____
_____	_____

What would you delete from the manual and why?

_____	_____
_____	_____
_____	_____

What areas need greater emphasis?

_____	_____
_____	_____
_____	_____

List any terms or symbols used incorrectly.

_____	_____
_____	_____
_____	_____

First Fold



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 00028 OREM, UTAH

POSTAGE WILL BE PAID BY ADDRESSEE

WICAT Systems, Inc.
Attn: Corporate Communications
1875 S. State St.
Orem, UT 84058

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



Second Fold

Tab