

SELCOPY

SELCOPY for AS/400, UNIX and PC Rel 2.08 – New Features

SELCOPY 2.08 New Features

SELCOPY for AS/400, UNIX and PC Rel 2.08 – New Features	1
Contents	2
Platforms.....	2
IMPORTANT CHANGES	4
DIR or DIRDATA input.....	4
.FIL as default file extension.....	4
DIFF, POS DIFF and POS UXADIFF.....	4
Continuation Records.....	5
PRINT TYPE=D format.....	6
NEW FEATURES	8
Release Information on Screen.....	8
Command line argument "-ctl fileid".....	8
Command line argument "-lst fileid".....	8
Command line argument "-log fileid".....	8
STDIN and STDOUT files.....	8
DIR and DIRDATA input for AS/400 and UNIX.....	9
SUBDIR parameter for DIR or DIRDATA input.....	9
SORTDIR parameter for DIR or DIRDATA input.....	10
POS VOLID for UNIX.....	10
POS RBA supported.....	10
KEY, RBA and REC for Direct Reads on a Disk File.....	11
STARTKEY, STARTRBA and STARTREC for Disk Files.....	12
OPT RC_KEYNF=n for the Key Not Found condition.....	12
OPT DEFDIR=path for I/O Operations.....	12
ASCII or EBCDIC Literals.....	13
POS UXREPLYL.....	13
POS UXLINE, UXLINEREM, UXPD, UXPW, UXDW.....	13
Floating Sign on FORMAT for CVxC.....	14
OTHER CHANGES	15
Expiry Date in Footing.....	15
Expiry Date Warning.....	15
Changes to Print output.....	15
Release Information in Footing.....	15
POS UXATPTR for 64-bit architectures.....	15
CARD instead of SYSIN.....	15
@ Ptr name length and total number.....	16
WORKLEN omitted with no input file.....	16
Max Len for BINary conversion.....	17
OPT ASA0 and NOASA0.....	17
ERROR Messages.....	17
Fixes in slc208.....	17

SELCOPY for AS/400, UNIX and PC Rel 2.08 – New Features

<http://www.cbl.com>

First published: **December 2002**

Contents

1. Platforms

- IMPORTANT CHANGES
 1. DIR or DIRDATA input
 2. .FIL as default file extension
 3. DIFF, POS DIFF and POS UXADIFF
 4. Continuation Records
 5. PRINT TYPE=D format
 - NEW FEATURES
 1. Release Information on Screen
 2. Command line argument "-ctl fileid"
 3. Command line argument "-lst fileid"
 4. Command line argument "-log fileid"
 5. STDIN and STDOUT files
 6. DIR and DIRDATA input for AS/400 and UNIX
 7. SUBDIR parameter for DIR or DIRDATA input
 8. SORTDIR parameter for DIR or DIRDATA input
 9. POS VOLID for UNIX
 10. POS RBA supported
 11. KEY, RBA and REC for Direct Reads on a Disk File
 12. STARTKEY, STARTRBA and STARTREC for Disk Files
 13. OPT RC_KEYNF=n for the Key Not Found condition
 14. OPT DEFDIR=path for I/O Operations
 15. ASCII or EBCDIC Literals
 16. POS UXREPLYL
 17. POS UXLINE, UXLINEREM, UXPD, UXPW, UXDW
 18. Floating Sign on FORMAT for CVxC
 - OTHER CHANGES
 1. Expiry Date in Footing
 2. Expiry Date Warning
 3. Changes to Print output
 4. Release Information in Footing
 5. POS UXATPTR for 64-bit architectures
 6. CARD instead of SYSIN
 7. @ Ptr name length and total number
 8. WORKLEN omitted with no input file
 9. Max Len for BINary conversion
 10. OPT ASA0 and NOASA0
 11. ERROR Messages
 12. Fixes in slc208
-

Platforms

The following platforms are supported by **SELCOPY**:

Note

This document does **NOT APPLY** to SELCOPY for mainframe.

CBL reserves the right to change, at its own discretion and without notice, the new features described in this document, until such time as incorporated in the official SELCOPY User Manual.

A complimentary hard copy of the user manual, documenting SELCOPY on all platforms, will be distributed to all customers when it becomes available.

The latest, published edition of the SELCOPY User Manual, which does not include these new features, may be viewed and downloaded in HTML format via CBL's web site.

IMPORTANT CHANGES

The changes documented here are considered "Important" because they may give rise to a noticeable difference in operational work.

Please therefore read the following headings, whose description follows immediately below:

DIR or DIRDATA input

(IMPORTANT)

For DIR or DIRDATA input, the default number of Sub Directory levels to be processed has been changed to 0.

UNIX systems are unaffected by this change as DIR and DIRDATA are new features in release 2.08 of UNIX SELCOPY. (See New Features below.)

SELCOPY releases prior to Rel 2.08 always nested down into all subdirectories, often to the irritation of the user.

Please see the heading below: "SUBDIR parameter for DIR or DIRDATA input" for controlling SELCOPY's DIR/DIRDATA sub directory processing.

.FIL as default file extension

(IMPORTANT)

UNIX systems are unaffected by this change.

For PC/DOS, OS/2 and Windows platforms, SELCOPY releases prior to 2.08 append a default file extension of **.FIL** to any input or output filename that is not prefixed by a disk letter or directory separator (the "\" or "/" character) and does not include a "." character.

Historically, this was intended to give compatibility with the mainframe operating system, VM/CMS, where a file extension is mandatory. However, the problems and confusion arising from it outweighed the benefit.

Filenames are now processed unchanged, as for UNIX.

DIFF, POS DIFF and POS UXADIFF

(IMPORTANT)

DIFF

The character string literal, 'DIFF', must be enclosed in quotes to avoid ERROR 134 being issued. e.g.

POS DIFF

Following a string compare, SELCOPY releases prior to 2.08 erroneously return the last byte of field1 for the "difference" position when field1 is shorter than field2 and field1 is equal to field2 for the full length of field1. i.e. When the difference found is in the padding of field1 and in the real data of field2.

It is of course totally illogical to return POS DIFF pointing to a character that was equal. Therefore, the position now returned is a virtual position in field1 representing the byte where the difference would have occurred if field1 had been the same length as field2.

When field1 is short, and is situated at the end of the workarea, then POS DIFF will point outside the workarea. Use of POS DIFF will then require a negative offset, bringing it back into the workarea, to allow access or modification of data.

The above change brings the AS/400, UNIX and PC versions of SELCOPY into line with the mainframe version for the setting of POS DIFF to a virtual position in the case of a short field1 with the difference only in the padding. However, it is still not identical.

The mainframe version of SELCOPY in these circumstances always sets POS DIFF to the virtual position representing the 1st byte of the padding, instead of at the virtual position in field1 representing where the difference would have occurred if field1 had been the same length as field2.

This is considered a bug in the mainframe version of SELCOPY which has now been fixed by Zap 54 (Ref: S200z54) available at <http://www.cbl.com>. This fix will be applied at source in the next release.

POS UXADIFF

The address pointer which may be referenced at POS UXADIFF is now held in Big Endian format in the same way as for POS UXATPTR, regardless of whether the host machine architecture is Big or Little Endian. Previous releases held UXADIFF in the local architecture.

Big Endian machines, SUN SPARC, IBM RS/6000 (p-series), HP PA-RISC and AS/400 for example, are unaffected by this change.

Continuation Records

(IMPORTANT)

The BackSlash character may be used as the last character on a control statement to indicate that a

continuation record follows on the next line. e.g.
becomes:

The limit of 512 bytes for the maximum length of a control statement still applies for release 2.08, even for a statement built using continuation records.

**** WARNING ****

Existing control statements should be checked for any occurrence of "\" as the last character of a record, because the next record will be treated as a continuation record, which could upset the logic.

It is recommended that all SELCOPY control statements are scanned for this possibility. e.g.

PRINT TYPE=D format

(IMPORTANT)

TYPE=D (Dump) printing now condenses multiple, consecutive lines of duplicate data into a single line with an indication of the number of lines condensed.

DUMPALL [= YES | NO]

Users post-processing such output, who may require that the format remains unchanged from previous releases, should use the DUMPALL parameter on the PRINT statement. DUMPALL with no arg is treated as DUMPALL=YES.

Alternatively, DUMPALL may be coded on an OPTION statement in the SELCOPY control statements, or as an installation standard in the selcopy.nam file.

DUMPENC = "xy" | "x"

The enclosing chars used by default for the character interpretation part of a TYPE=D print line has changed from "*" to "|" for better readability, but may be changed as required by the user with the DUMPENC parameter on an individual PRINT TY=D statement, or on an OPTION statement.

The DUMPENC argument may only be 1 or 2 characters. Default is DUMPENC="|":

The 1st char defines the enclosing character for the character interpretation part of a TYPE=D line. The 2nd char defines the enclosing character for the duplicate TYPE=D data lines which get suppressed into a single "=same=" line on the output, provided DUMPALL has not been coded on the PRINT or OPT statement. e.g.

NEW FEATURES

Release Information on Screen

The command line option `-V` (uppercase) has been introduced to give information on the platform, release number and build level directly on screen.

The release information is written to `stderr` and SELCOPY terminates without any further processing, regardless of other arguments or input redirection.

Redirection of `stderr` is respected. e.g.

will produce the following:

Command line argument "`-ctl fileid`"

By default, SELCOPY reads its control statements from `stdin`. This may now be overridden by use of the command line argument "`-ctl fileid`" (without the quotes), where `fileid` is the name of the control statement file prefixed with a relative or absolute path as required.

**** IMPORTANT ****

This then releases `stdin` for use as a regular input file, allowing use of SELCOPY in a standard pipe.

Command line argument "`-lst fileid`"

By default, SELCOPY writes its PRINT output to the file "SELC.LST" on the current directory, or to the `[path]filename` specified by the environment variable `SLCLST`. This may now be overridden by use of the command line argument "`-lst fileid`" (without the quotes), where `fname` is the name of the required listing file prefixed with a relative or absolute path as required.

Command line argument "`-log fileid`"

By default, SELCOPY writes its LOG output, and certain error information, to `stderr`, the system standard error file. This may now be overridden by use of the command line argument "`-log fileid`" (without the quotes), where `fname` is the name of the required output file prefixed with a relative or absolute path as required.

STDIN and STDOUT files

The filenames "`stdin`" and "`stdout`" are now recognized by SELCOPY as standard predefined system files and processed as such.

SELCOPY for AS/400, UNIX and PC Rel 2.08 – New Features

The command line argument "-ctl fileid" must be used to define the name of the control statement input file, thereby freeing up stdin for use as a regular input file. e.g.

Suppose we have a file called test.ctl on the current directory containing the 4 lines:

Then we issue the command:

The output (stdout) from the dir command will be piped into SELCOPY where it is read as stdin, but only the 1st 22 lines of it are written to stdout, which is not piped, so goes to the terminal. The 1st 11 lines are also printed on the SELC.LST file, which will also report on how many stdin records were processed.

The command:

would achieve the same result without the requirement to have a real file for the control statements.

By default, "stdin" and "stdout" are buffered by SELCOPY, using the default buffer size of 2048. This default may be overridden with the B=nnn (or BLKSIZE=nnn) parameter on the READ or WRITE statement.

As for normal files, the default record format is RECFM=U (undefined length records) terminated with LF or CRLF according to the host operating system. RECFM=x and LRECL=n may be coded to override this.

DIR and DIRDATA input for AS/400 and UNIX

Now supported, with timestamp, filename and filesize information reported in the same way as on the PC platforms. Please see "POS VOLID for UNIX" below for example of usage.

Positions 22–27 of the DIR record contain file information flags as obtained following a **stat** function. The meanings of these flags differ between UNIX systems and are, therefore, not documented here. Please refer to the **include/sys/stat.h** file on your UNIX system for a full description of each flag.

Flags that are common to all UNIX systems are the file permissions which are reported in the traditional octal format in positions 25–27. e.g.

As on the PC platforms, the SUBDIR parameter allows nesting into sub-directories.

SUBDIR parameter for DIR or DIRDATA input

By default, the number of Sub Directory levels processed for DIR or DIRDATA input is 0, but the SUBDIR parameter, with an optional numeric argument, may be used to control this.

SUBDIR may be coded either on the READ statement for the file in question, or on an OPTION statement which will affect all DIR and DIRDATA input.

SUBDIR may optionally be coded with a numeric argument to define the number of directory levels into which SELCOPY DIR/DIRDATA processing will nest. Coding SUBDIR without a numeric argument results in SUBDIR=256 being assumed.

When SUBDIR is coded on an OPTION statement in the selcopy.nam file, it will apply to all DIR and DIRDATA input, as an installation standard default, for all SELCOPY jobs.

SUB is a synonym for SUBDIR, and NOSUB is a synonym for SUBDIR=0. e.g.

SORTDIR parameter for DIR or DIRDATA input

For DIR and DIRDATA input, SORTDIR=x may be coded to request that DIR records returned are sorted by "x" where "x" is:

By default, all DIR and DIRDATA input is returned unsorted. i.e. in the order returned from the operating system.

The SORTDIR parameter may be coded on the READ statement for the file in question, or on an OPTION statement which will affect all DIR and DIRDATA input.

Use the OPTION statement in the selcopy.nam file to affect all DIR and DIRDATA input as an installation standard. Coding SORTDIR and its argument within SELCOPY control statements will override the SORTDIR argument set in selcopy.nam. e.g. SORTDIR=NO may be coded on a file's READ statement to get unsorted input for that file only, thus overriding the installation standard in the selcopy.nam file.

Note that both Size and Date options will return the DIR entries in descending order.

SORT is a synonym for SORTDIR, and NOSORT is a synonym for for SORT=NO (unsorted). e.g.

POS VOLID for UNIX

For PC platforms, POS VOLID refers to a 32 byte, blank padded or truncated, field containing the Volume Id of the hard disk holding the file last processed.

For UNIX, this is inappropriate, and the hostname (nodename) of the local machine is supplied at POS VOLID instead.

Note that the volid is not automatically included as part of a DIR record. It should be moved into the appropriate position as required. e.g.

```
selcopy "-lst x.x ! in '/usr/cbl/djh/*' dir w 99 ! if p 30 = '!' ! t move 6 at volid to 43 ! t pr dw=90 !e"
```

produces the following in the file "./x.x", where "s01" is the hostname:

POS RBA supported

For all disk file I/O, the **current** RBA (Relative Byte Address) of the last file read or written, is made available to the user via **POS RBA**.

For **input**, the current RBA is the relative byte address of the last logical record read. RBA=0 can therefore mean either, nothing has yet been read, or only 1 record has been read.

For **output**, the current RBA is the relative byte address of the **next** logical record to be written, **not** the last record written. Thus RBA=0 for an output file means that nothing has yet been written.

POS RBA refers to the junior 4-bytes of an 8-byte RBA on all systems. Where the operating system supports file sizes in excess of 4 gigabytes, the senior 4-bytes will be non-zero as appropriate. e.g.

For input, this can provide a useful mechanism for saving the current RBA, continuing processing on the file, then going back to the saved file position with a direct read by RBA. See below.

For output, the RBA of the next record to be written can be included as part of the data in the output record.

KEY, RBA and REC for Direct Reads on a Disk File

The following syntax may be used for direct reads into the middle of a disk file:

Provided a key field exists in every record of a disk file, and the key field is in ascending sequence, with no duplicates, any record may be read directly by quoting the required key.

The LRECL (L), KEYLEN (KL) and KEYPOS (KP) parameters are all mandatory. The RKP (Relative Key Pos) parameter may be used instead of KEYPOS. (RKP=0 means KP=1.)

Any disk file may be read by specifying the required RBA (Relative Byte Address) of the file to read the record. May be used on any RECFM.

The buffer will be filled from the RBA point in the file, and the first full record following that will be returned, ignoring any partial record that might exist at that RBA. The RBA value set at POS RBA will then be that of the record returned, which may well be greater than the RBA requested.

The KEYLEN and KEYPOS parameters should only be coded for READ by RBA if the file does indeed have a "key" field and is RECFM=V, in which case the information will be used to improve RDW validation.

On RECFM=F files only, any disk file may be read by specifying the required record number. RECFM may be omitted because RECFM=F is the default when L=n is coded.

Notes on Direct Reads:

1. For KEY and RBA reads on a RECFM=V file, an LRECL=n value is mandatory in order to give SELCOPY a maximum value to improve on its RDW validation. Then, because LRECL=n is coded to define the maximum LRECL, a RECFM definition is required to override the default of RECFM=F which is implied by having LRECL=n coded.
2. For all RECFM=V, it is recommended that buffer size be at least double the LRECL. This is to improve RDW validation. However, it should be noted that use of a large buffer size causes additional unnecessary I/O for direct reading.
3. For all RECFM=V input, the KEYPOS or RKP (Relative Key Pos) parameters are based on the position or offset within the data portion of the record, totally disregarding the RDW which always exists, and also totally disregarding whether the RDW or NORDW option is in effect. Thus, the KEYPOS (or RKP) parameters do not require to be changed if the file is copied to a file of different record format, or if the user changes the NORDW option to RDW. For UNIX and PC platforms, NORDW is the default. For mainframe platforms, RDW is the default.
4. The NOBDW option may be coded when it is known that the input file does not have a Block Descriptor Word at the start of each logical block, simulating a mainframe RECFM=V file. By default, it is assumed that valid BDWs exist and logical records within a block will be processed individually. If NOBDW is coded, the check for a valid BDW will be bypassed, so if BDWs do actually exist, then each BDW encountered will cause that complete block to be processed as an individual record, which on a keyed file (KP=n and KL=n coded) would almost certainly result in a Key Sequence Error.
5. Direct READ statements may be combined with normal READ statements. If after a direct READ, a normal sequential READ statement is actioned for a file, the next sequential record is returned. i.e. The direct read will leave the file positioned correctly following the record read. Thus, EOF would be returned for a sequential read if the preceding direct read gave the "---- KEY NOT FOUND ----"

condition after a request to read at or beyond the filesize.

STARTKEY, STARTRBA and STARTREC for Disk Files

The following syntax may be used for starting sequential input from the middle of a disk file:

Syntax is as defined for KEY, RBA and REC parameters of the READ statement. Input will commence at the record with the specified KEY, RBA or REC value, continuing thereafter with the next sequential record as though it were a normal read.

OPT RC_KEYNF=n for the Key Not Found condition

When a direct read gives the "---- KEY NOT FOUND ----" condition after a request to read a non-existent record or beyond the filesize, by default, conforming to the existing mainframe version of SELCOPY, no return code is set, and nothing is reported in the summary for that selection.

The RC_KEYNF=n option in selcopy.nam, or on the SELCOPY control statements, can be coded to define the required return code for the Key Not Found condition, where n is from 0 to 254. e.g.

If the argument of the RC_KEYNF option is non-zero, the RC will be set as requested, however, the event will be considered as a RC=8 for the purpose of reporting in the summary for that selection.

OPT DEFDIR=path for I/O Operations

A default runtime directory may be defined with the DEFDIR parameter, synonyms DEFAULTDIR and DFLTDIR, on an OPTION statement. e.g.

OPTION DEFDIR=path may be coded in the selcopy.nam file, in the SELCOPY control statements, or in both. The last DEFDIR coded must appear before any file I/O statements.

When DEFDIR=xxx has been coded, and a file I/O statement references a DSN, or filename without a DSN, which does not start at the root directory of a filesystem, then the DEFDIR path is used as a prefix instead of the system's current working directory.

If the DEFDIR argument is a relative path, it is appended to the system's current working directory.

If the DEFDIR argument is an EQU name, then the argument of the EQU statement should be enclosed in quotes in order to prevent the string being uppercased during the EQU processing.

System environment variables may be used as part or all of the DEFDIR argument. No change is made to the system's current working directory.

ERROR 169 will be issued if an OPTION DEFDIR is encountered following any file I/O statement, regardless of whether the existing DEFDIR path was specified.

ASCII or EBCDIC Literals

The **ASC** and **EBC** parameters are supported on any operation that uses literal strings.

For SELCOPY Rel 2.08 BuildLevel=345 and above, the synonyms **ASCII** and **EBCDIC** are also supported. (You can check the build level with the command: "selcopy -V".)

ASC or EBC may be coded after a **quoted literal** to indicate that the literal is to be treated as being in the specified ASCII or EBCDIC coding convention, regardless of the native coding convention of the host machine. e.g.

By default, literals are stored in the native coding convention of the host machine.

POS UXREPLYL

The 4-byte binary length of the last reply given by the operator in response to a LOG message with REPLY parameter, may now be referenced at POS UXREPLYL.

POS UXLINE, UXLINEREM, UXP, UXPW, UXDW

POS UXLINE

has been fixed to give same information on the number of lines already printed on the current page, as does the mainframe version of SELCOPY.

POS UXLINEREM

has been introduced, referencing a 4-byte binary field containing the number of lines remaining on the current page.

POS UXP

has been introduced, referencing a 4-byte binary field containing the current page depth.

POS UXPW

has been introduced, referencing a 4–byte binary field containing the current page width, as set by the PAGEWIDTH option.

POS UXDW

has been introduced, referencing a 4–byte binary field containing the current data width, as set by the DATAWIDTH option.

Floating Sign on FORMAT for CVxC

The 'S' format character, (synonyms: 's' and '+'), of any CVxC conversion statement, may be used repeatedly to indicate that the resultant sign is to be floated down to the junior insignificant 'S', and always displayed as either '+' or '-'.

The 'S' format character is processed as a 'Z' format character, giving zero suppression, except that the junior insignificant 'S' is replaced with the sign of the source. e.g.

The '-' format character is similar to the 'S' format character, except that the junior insignificant '-' is only replaced with the sign of the source when negative. When the source is positive, the '-' format character is replaced with a significant digit or blank. e.g.

'S' or '-' format characters should not be used following 'Z' or '9' format characters, because the sign position could get used for a significant digit. An exception is made for a '-' format character which is immediately preceded by a '9' format character. In this case, the '-' is processed as a punctuation character. However, use of '-' as a punctuation character is best avoided. e.g.

'S' or '-' format characters may be followed by 'Z' format characters, giving further zero suppression, but the user should beware of losing what could be important punctuation, such as a decimal point, when zero suppression is still active. This has always been a danger, even with the traditional ZZZ method. e.g.

Important punctuation should always be preceded by a '9' in the FORMAT argument.

In all cases, non–format data to the right of the result is processed in the traditional way, getting suppressed when the source is not negative. e.g.

OTHER CHANGES

Expiry Date in Footing

The correct expiry date is now reported in the footing line of the SELCOPY list output.

Expiry Date Warning

When running within 28 days of the expiry date, SELCOPY will issue a warning message on the screen. e.g.

Changes to Print output

The counting guide at the bottom of each page uses:

instead of:

Changes to the footing lines are discussed below.

Release Information in Footing

The format of the footing has been adjusted to give information on the platform, release number and build level.

Mixed case is used in the 2 footing lines which previously were fully upper case. e.g.

instead of:

POS UXATPTR for 64-bit architectures

This change only applies to HP's Tru64 UNIX.

For compatibility with 32-bit platforms, and in keeping with all other 8-byte variables made available to the SELCOPY user on a 64-bit machine, POS UXATPTR refers to the first of the junior 4 bytes of the 8 byte address.

Releases prior to SELCOPY 2.08, gave POS UXATPTR as the first byte (senior) of the 8 byte address, which gave porting problems.

CARD instead of SYSIN

SELCOPY for AS/400, UNIX and PC Rel 2.08 – New Features

The statement "READ FILE=CARD" was previously reported in the summary with a filename of "SYSIN". This is no longer the case. "READ FILE=CARD" is now reported in the summary with a filename of "CARD".

The statement "READ FILE=SYSIN" is no longer synonymous with "READ FILE=CARD". It is now reported in the summary with a filename of "stdin", and treated as such.

READ FILE=CARD

READ CARD will read a record from the file from which the SELCOPY control statements were read.

By default, standard system input (stdin) is used for SELCOPY's control statements, which may be piped or redirected.

However, use of the "-ctl fileid" command line syntax, (see New Features), will allow the stdin default to be overridden. Control statements would then be read from the -ctl argument, leaving stdin available for use as a separate file with its piping or redirection capability.

An END statement must have been present in order to indicate the end of the SELCOPY control statements. Records following the END statement will then be used as input to the file, CARD, when a "READ FILE=CARD" statement is actioned.

READ FILE=SYSIN

SYSIN is treated as meaning standard system input, (and is synonymous with READ FILE=STDIN), regardless of what file was used to read SELCOPY's control statements.

When the statement "READ FILE=SYSIN" is actioned, a record will be read from stdin. If stdin is not redirected or piped, a record will be read from the terminal.

@ Ptr name length and total number

The restriction of length 4 for an @ pointer name has been removed as has the restriction of no more than 32 @ pointers.

Thus, the following error messages will no longer occur on the AS/400, UNIX or PC platforms.

However, it should be noted that these restrictions still apply on SELCOPY release 2.00 for mainframe.

WORKLEN omitted with no input file

If WORKLEN is omitted, and no input file is used, a default record area of 80 blanks is provided by default. Thus, the control statements in the example below are acceptable.

SELCOPY releases prior to Rel 2.08 would give RC=8 for all statements used in the example.

Max Len for BINary conversion

The maximum length allowed for a CVBx statement has been increased from 4 to 8.

OPT ASA0 and NOASA0

For compatibility with SELCOPY mainframe, the options ASA0 and NOASA0 are silently ignored by the AS/400, UNIX and PC versions.

ERROR Messages

Selection Summary Messages

New message text:

**NOT*FOUND*OR*EMPTY ###**

replaces existing message:

**FILE*NOT*FOUND ###**

Fixes in slc208

INC statement

The INCLUDE statement no longer gives ERROR 156 when a comment exists on the control statement, and now supports filenames with embedded blanks.

POS UXADIFF

POS UXADIFF is now correctly presented in Big Endian format as a 4–byte address. Previous releases gave Big or Little Endian format depending on the local machine architecture.

POS UXATPTR

POS UXATPTR now points to a 4–byte address, regardless of machine architecture. On a 64–bit machine, the user may refer to POS UXATPTR–4 for the full 8–byte address.

Default LRECL

(Build Level=159 2002/08/24 12:50)

When RECFM and LRECL are both omitted for an output file, both attributes are defaulted to the same as those of the Prime Input file. Previous releases only defaulted the RECFM to the same as the Prime Input file, which then caused LRECL to default to buffersize. This was particularly undesirable for RECFM=F.

Nulls in DIR record

(Build Level=160 2002/08/26 20:14)

For Windows NT only, the volid displayed in a DIR record contained x'00' chars if the volid had less than 11 characters.

@ Pointer Literals

(Build Level=294 2003/02/10 16:33)

Arithmetic operations using an @ pointer or LRECL as a literal value for TYPE=P fields will no longer give RC=8. e.g.

Keyed Reads

(Build Level=309 2003/06/02 18:26)

RECFM=F input with a buffer size not a multiple of LRECL gave ERROR 537 (Not in Key Seq) erroneously. Files with less than 3 records also gave ERROR 537.

A keyed read on an empty or non-existent file should return "Key not found" instead of terminating with "Empty File" error.

Keyed Reads

(Build Level=332 2003/07/12 21:56)

Where the buffer size is equal to or greater than filesize, only 1 physical read should be required.