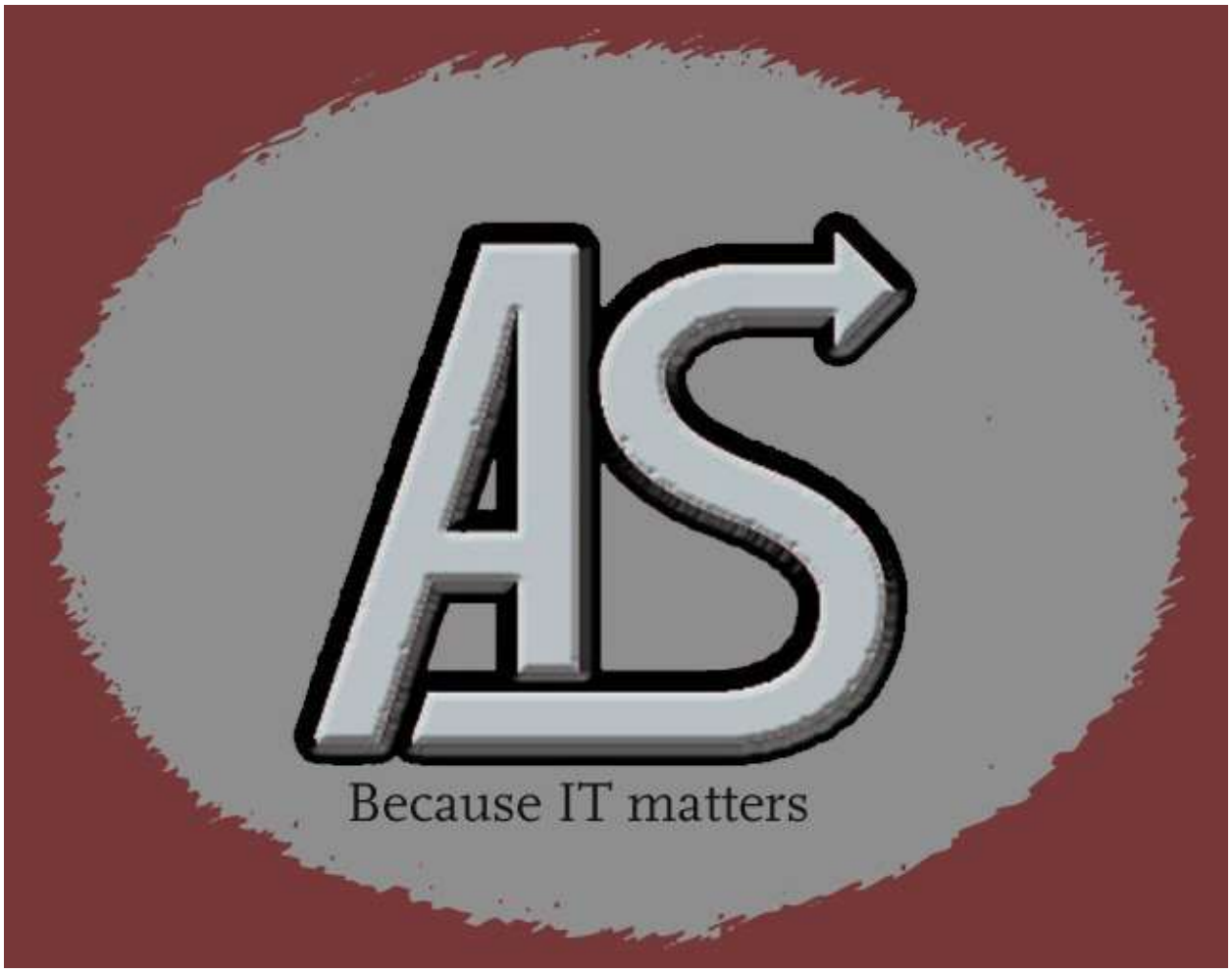
Abbydale Systems LLC.



ASLCOPY

Program Information

E-mail: Support@abbydalesystems.com

C O P Y R I G H T

Abbydale Systems LLC.
2925 Gulf Freeway South
Suite B #229
LEAGUE CITY
Texas USA
77573

ATTN: K.E.Ferguson
Legal@abbydalesystems.com

Disclaimer

This computer program and associated materials was developed by Kevin E. Ferguson of **Abbydale Systems LLC.**

This material has been used successfully by **Abbydale Systems LLC.** and to the best of our knowledge this material and any system(s) of which it is a part are operational as of the service level or date stated in the body of this material (if so stated). However, ***no warranty*** is given or implied as to the accuracy of this material or any related material or systems, and ***no responsibility*** is assumed for any effect or modification directly or indirectly caused by the use of this material.

It is the responsibility of any user of this material to evaluate its usefulness to the user's environment.

Abbydale Systems LLC. does not guarantee to keep this nor any related material current, nor does it guarantee to provide any corrections or extensions described by any users of this material or any corrections or extensions made in the future by **Abbydale Systems LLC.** itself.

Acknowledgements

This document refers to several software products that are produced by other companies. In most cases the names of these products are trademarks and/or copyright of those companies. It is not our intention to claim either the name of the trademark, nor the product itself, these remain solely the right of the owning companies.

CONTENTS

1 Overview

The program, **ASLCOPY**, is used as an IEBCOPY front end filter.

It will generate IEBCOPY control cards for a partitioned dataset (PDS) based on the parameter list passed to it.

ASLCOPY can be used to filter which members of a PDS are included or excluded from being copied based on who last updated the member and/or generic name of the members. ALTERMOD cards can also be generated for members of a load library.

1.1 Passed Parameters

ASLCOPY must have a parameter list passed. If no parameters are passed the program will terminate with a condition code of 12 or a S0C3 abend depending on the presence of a NOABEND DD card.

The passed parameter list is **not** case sensitive. It will be folded into uppercase by **ASLCOPY**.

Note: ASLCOPY parameters are positional, if they are not specified, they should still provide a comma to indicate an unused parameter.

Note: You should only need to use ALTERMOD on modules that were written by a linkage editor prior to MVS/370 or if you believe that the RLD records are corrupt for a load module.

The parameter list for **ASLCOPY** is as follows:

Action	Specify either SELECT, EXCLUDE, or ALTERMOD to determine what type of IEBCOPY statements are to be generated. **ALTERMOD is only available for load libraries**.
R	Specifies whether the REPLACE format of IEBCOPY control cards will be generated. **The replace parameter is not available with EXCLUDE**
BY=	Specifies the userid (or generic userid) of the users who last updated members of the PDS for inclusion or exclusion. There are 2 special values allowed for BY= these are *ALL* This instructs **ASLCOPY** to process all members that were last updated by anyone. *NONE* This instructs **ASLCOPY** to process only members that have not been updated by anyone. **This option is not available for load libraries.**

Mem,mem2	Specifies the member names to be included or excluded

1.2 Parameter Examples

If you want to copy all members that start with ABC from a PDS library without replacing existing members in the output dataset then you would code:

```
//STEP EXEC PGM=ASLCOPY,PARM=(SELECT,,,ABC)
```

To do the same thing but replace existing output dataset members you would code:

```
//STEP EXEC PGM=ASLCOPY,PARM=(SELECT,R,,ABC)
```

If you want to copy and replace all members from a PDS that were last updated by USER1 then you would code:

```
//STEP EXEC PGM=ASLCOPY,PARM=(SELECT,R,BY=USER1)
```

If you want to exclude all members of a load library with module names starting IEB then you would code:

```
//STEP EXEC PGM=ASLCOPY,PARM=(EXCLUDE,,,IEB)
```

In this example everything in the load library other than modules named IEBxxx would be copied.

If you want to copy and re-block a load modules for all ASL modules you would code:

```
//STEP EXEC PGM=ASLCOPY,PARM=(ALTERMOD,,,ASL)
```

ASLCOPY gives the user the option to either ABEND with a S0C3 or issue a condition code of 12 if it encounters and error in the parameter list. This is controlled by the use of a NOABEND DD DUMMY card.

The default is for **ASLCOPY** to terminate with an S0C3.

1.3 Called Programs: *None* ***(Indirectly calls)*** *None*

1.4 IBM macros used:

CAMLST	CLOSE	DCB	GET
IECSDSL1	IEFJFCBN	OBTAIN	OPEN
PUT	RETURN	RDJFCB	WTO

1.5 User macros used:

EYECATCH

1.6 Assembled User Values

ASLCOPY is distributed as an Object Code Only (OCO) program so there are no user values that can be assembled.

2 Installation Procedure

2.1 From XMI File

The XMI (or XMIT) file is in IBM TSO TRANSMIT format and **must** be transferred to z/OS™ as a fixed blocked 80 byte BINARY file. The disk space requirement for the file is 2 tracks of 3390 disk.

The FTP process (if performed in a 3270 emulator) must be performed in TSO READY mode or in option 6 of ISPF™.

The dataset name used as input for the TSO TRANSMIT was ABBYDALE.ASLCOPY.PDS. Unless this is changed by the TSO RECEIVE command it will be the name of the dataset created by the RECEIVE command.

3 Using ASLCOPY

The program. **ASLCOPY,** has no other function than to write out the contents of the passed parameter into an output dataset as defined on the STEPCRD DD card.

It will, if required, add a continuation character into column 72 and it can also fold the parameter string into upper case if desired.

3.1 Required JCL for ASLCOPY

```
1   //Jobname JOB ……..
2   //stepname EXEC PGM=ASLCOPY,PARM='parm'
3   //STEPLIB  DD DSN=your.load.library,DISP=SHR
4   //INPDS    DD DISP=SHR,DSN=input.pds
5   //SYSPRINT DD SYSOUT=*
6   //NOABEND  DD DUMMY
7   //STEPCRD  DD UNIT=SYSDA,
    //         SPACE=(TRK,(10,1)),DISP=(,PASS)
```

JCL card	Required	Use
1	Yes	JOBCARD (Tailor to your site standards)
2	Yes	EXEC card. A parameter list **must** be passed to this program.
3	Yes	This is the library where **ASLCOPY** is located The STEPLIB concatenation can only be removed if the module resides in LNKLST libraries.
4	Yes	The INPDS DD card specifies the input partitioned dataset that the members to be copied reside in.
5	No	The SYSPRINT DD card is where any error messages are written. If the SYSPRINT card is missing, then the error messages are written to the console via a WTO.
6	No	The NOABEND DD card, if present, cause the program to issue a return code of 12 if it encounters any errors. If the card is missing, then the program will abend with a S0C3 if it encounters an error.
7	Yes	The STEPCRD is where **ASLCOPY** will write the generated IEBCOPY control statements.

4 Sample JCL

4.1 Backup All Members Changed by a Particular User

In the sample job **ASLCOPY** is used to generate the required IEBCOPY control cards to copy all the members of the partitioned dataset ABBYDALE.TEST.PDS that were last updated by MIT001 to a new partitioned dataset named ABBYDALE.TEST.PDS.BKUP.

The first step, STEP1, uses program **ASLDEL** to delete the backup dataset (if it exists 0)

The second step, STEP2, executes **ASLCOPY** to generate IEBCOPY SELECT statements for the members in ABBYDALE.TEST.PDS that were last updated by the user MIT001.

The third and final step, STEP3, uses IEBCOPY to copy the member list generated by **ASLCOPY** from ABBYDALE.TEST.PDS to ABBYDALE.TEST.PDS.BKUP

```
//ASLCOPY JOB
//         SET PDS=ABBYDALE.TEST.PDS
//STEP1    EXEC PGM=ASLDEL,PARM='RC=0,&PDS..BKUP'
//STEP2    EXEC PGM=ASLCOPY,PARM='SELECT,,BY=mit001'
//STEPLIB  DD   DISP=SHR,DSN=ABBYDALE.DEVL.LOADLIB
//INPDS    DD   DISP=SHR,DSN=&PDS
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//STEPCRD  DD   UNIT=SYSDA,SPACE=(TRK,(10,1)),DISP=(,PASS)
//STEP3    EXEC PGM=IEBCOPY,COND=(0,NE)
//SYSPRINT DD   SYSOUT=*
//SYSUT3   DD   SPACE=(80,(30,5)),UNIT=SYSDA
//INPUT    DD   DSN=*.STEP2.INPDS,DISP=SHR
//OUTPUT   DD   DSN=&PDS..BKUP,
//         SPACE=(CYL,(55,4,900),RLSE),
//         VOL=SER=ABBY02,UNIT=SYSALLDA,DISP=(,CATLG),
//         DCB=&PDS
//SYSIN    DD   DSN=*.STEP2.STEPCRD,DISP=SHR
```

4.2 Backup All Members With Names Starting With ASL

In the sample job **ASLCOPY** is used to generate the required IEBCOPY control cards to copy all the members of the partitioned dataset ABBYDALE.TEST.PDS whose names start with ASL to a new partitioned dataset named ABBYDALE.TEST.PDS.BKUP.

The first step, STEP1, uses program **ASLDEL** to delete the backup dataset (if it exists 0)

The second step, STEP2, executes **ASLCOPY** to generate IEBCOPY SELECT statements for the members in ABBYDALE.TEST.PDS that start with the characters ASL.

The third and final step, STEP3, uses IEBCOPY to copy the member list generated by **ASLCOPY** from ABBYDALE.TEST.PDS to ABBYDALE.TEST.PDS.BKUP

```
//ASLCOPY JOB
//         SET PDS=ABBYDALE.TEST.PDS
//STEP1    EXEC PGM=ASLDEL,PARM='RC=0,&PDS..BKUP'
//STEP2    EXEC PGM=ASLCOPY,PARM='SELECT,,,ASL'
//STEPLIB  DD   DISP=SHR,DSN=ABBYDALE.DEVL.LOADLIB
//INPDS    DD   DISP=SHR,DSN=&PDS
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//STEPCRD  DD   UNIT=SYSDA,SPACE=(TRK,(10,1)),DISP=(,PASS)
//STEP3    EXEC PGM=IEBCOPY,COND=(0,NE)
//SYSPRINT DD   SYSOUT=*
//SYSUT3   DD   SPACE=(80,(30,5)),UNIT=SYSDA
//INPUT    DD   DSN=*.STEP2.INPDS,DISP=SHR
//OUTPUT   DD   DSN=&PDS..BKUP,
//         SPACE=(CYL,(55,4,900),RLSE),
//         VOL=SER=ABBY02,UNIT=SYSALLDA,DISP=(,CATLG),
//         DCB=&PDS
//SYSIN    DD   DSN=*.STEP2.STEPCRD,DISP=SHR
```

4.3 ALTERMOD All Members of a Load Library Starting with ABC

In the sample job **ASLCOPY** is used to generate the required IEBCOPY control cards to alter the load modules in place. This process is only really need if the load modules were built using a linkage editor prior to MVS/370 but it may also be required to rebuild the RLD of the module.

The first step executes **ASLCOPY** to generate IEBCOPY ALTERMOD statements for the members in ABBYDALE.TEST.PDS that start with the characters ABC.

The second and final step uses IEBCOPY to alter the load modules

```
//ASLCOPY JOB
//         SET PDS=ABBYDALE.TEST.LOADLIB
//STEP1    EXEC PGM=ASLCOPY,PARM='ALTERMOD,,,ABC'
//STEPLIB  DD   DISP=SHR,DSN=ABBYDALE.DEVL.LOADLIB
//INPDS    DD   DISP=SHR,DSN=&PDS
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//STEPCRD  DD   UNIT=SYSDA,SPACE=(TRK,(10,1)),DISP=(,PASS)
//STEP2    EXEC PGM=IEBCOPY,COND=(0,NE)
//SYSPRINT DD   SYSOUT=*
//SYSUT3   DD   SPACE=(80,(30,5)),UNIT=SYSDA
//OUTPUT   DD   DSN=*.STEP1.INPDS,DISP=SHR
//SYSIN    DD   DSN=*.STEP1.STEPCRD,DISP=SHR
```

5 Messages

ASLCPY01E

ASLCPY01E – No generic member name passed.

Meaning

No parameter was passed to the program.

Corrective Action

Correct the error and rerun as required.

ASLCPY02E

ASLCPY02E – SELECT, EXCLUDE or ALTERMOD must be specified

Meaning

The passed parameter list does not specify SELECT, EXCLUDE or ALTERMOD as the first parameter

Corrective Action

The parameter list should be corrected, and the job rerun as required. Please refer to the “Passed Parameters” section of this document for the required order of the parameter list.

ASLCPY03E

ASLCPY03E – Only R may be specified on a replace card.

Meaning

The program detected that something other than an R was specified for replace. The replace parameter is positional and if it is not used a comma should be coded where the R would have been.

Corrective Action

Correct the error and rerun if required.

ASLCPY04E

ASLCPY04E – Invalid BY= field.

Meaning

The program detected that the BY= field is invalid. This is usually that the field is too long but could also be that the USERID passed contains invalid characters.

Corrective Action

Correct the BY= value and rerun the job.

ASLCPY05E

ASLCPY05E – No members specified for copy.

Meaning

The program detected that there was no member name stem specified for the execution of the program. The only circumstance where no member name stem needs to be specified is when using the BY= option.

Corrective Action

Correct the error and rerun if required.

ASLCPY06E

ASLCPY06E – Replace and ALTERMOD are mutually exclusive.

Meaning

The program detected that the BY= field is invalid. This is usually that the field is too long but could also be that the USERID passed contains invalid characters.

Corrective Action

Correct the BY= value and rerun the job.

ASLCPY07E

ASLCPY07E – Invalid parameter format.

Meaning

The program has detected an error in the passed parameter format

Corrective Action

Correct the error and rerun if required.

ASLCPY08E

ASLCPY08E – OBTAIN Failed.

Meaning

The program detected an error when issuing the OBTAIN macro for the input PDS. This could mean that the partition dataset directory is corrupt.

Corrective Action

Investigate the partitioned dataset and correct any issues.

If the problem persists, please contact Abbydale Systems for technical support. We will need a SYSUDUMP of the failing job and may request a dump of the PDS directory.

ASLCPY09E

ASLCPY09E – Member name length error.

Meaning

The program detected that there was no member name stem specified for the execution of the program. The only circumstance where no member name stem needs to be specified is when using BY=

Corrective Action

Correct the error and rerun if required.

ASLCPY10E

ASLCPY10E –ALTERMOD only valid for load libraries.

Meaning

The program detected that the BY= field is invalid. This is usually that the field is too long but could also be that the USERID passed contains invalid characters.

Corrective Action

Correct the BY= value and rerun the job.

ASLCPY11E

ASLCPY11E – BY= processing not allowed on load libraries.

Meaning

The program detected that there was no member name stem specified for the execution of the program. The only circumstance where no member name stem needs to be specified is when using BY=

Corrective Action

Correct the error and rerun if required.

ASLCPY12E

ASLCPY12E – No parameter list passed

Meaning

The program was not passed any parameter list and can't continue.

Corrective Action

Provide a valid parameter list to the program and rerun if required.

ASLCPY13E

ASLCPY13E - Storage obtain failed. Abend S0C3

Meaning

The was unable to obtain storage for the program.

Corrective Action

Increase the region size for the job step and rerun the job. If the problem persists, please contact Abbydale Systems LLC for support if required.

ASLCPY99E

ASLCPY99E – Dataset not found

Meaning

The OPEN for the input partitioned dataset failed.

Corrective Action

Check that the dataset defined on the INPDS JCL card exists. Correct the error and rerun the job if required.

6 Summary of Amendments

Date	Version	Fix Id.	Comment
8th March 2022	3.2	n/a	Release version.
3rd August 2017	3.1	n/a	Fix for allowing WTO instead of writing to SYSPRINT
28th May 2017	2.1	n/a	Ownership transferred to Abbydale Systems LLC.
16th June 1997	3.1	n/a	Code rewritten as UTCOPY for Bowstore Ltd.
20th February 1988	2.1	n/a	Code to support BY= and Replace.
23rd April 1987	1.0	n/a	Initial program written

Obtaining Support

Support for, comments about and suggestions for enhancements for this product can be obtained from our website :

www.abbydalesystems.com

or by emailing us at

support@abbydalesystems.com

In order to assist us in filtering support emails please specify in the heading of the email the name of the product that you require support on.

Spam will not be tolerated at this email address.

Where source code is provided for the product, support will be on a 'best efforts' basis. Where the user site has modified the source code, support may entail requesting copies of that sites source code and may result in support being withdrawn if this is not provided.

Abbydale Systems LLC. reserves the right to any code modifications that may have been undertaken at the user site.

Any alteration of the copyright information contained in the original source code is an infringement of the copyright of this and any other Abbydale Systems product and may result in legal action being taken against the perpetrator.