



CSI

INTERNATIONAL

Original Printing 01/22/2003
Last Revised..... 01/22/2003

Table of Contents

PRODUCT OVERVIEW	5
Introduction	5
Features and Capabilities	6
INSTALLATION	7
Load Distribution Tape	7
Option Parameters Default Modifications	8
HOW JCL/ANALYZER WORKS	9
Supported Environments	9
System Requirements	9
Functional Flow	9
Member Retrieval	9
CONTROL STATEMENTS	11
Control Statement Coding Rules	11
Function Keywords	12
Option Parameters	14
Selection Parameters	18
Selection Parameter Coding Rules	18
Include Parameters	18
Exclude Parameters	20
EXAMPLES OF USE	21
JOB CONTROL LANGUAGE	22
Job Control Statements	22
Sample Job Execution	26
REPORT EXAMPLES	27
JOB Cross-reference Report	27
PROCEDURE Cross-reference Report	28
PROGRAM Cross-reference Report	29
FILE Cross-reference Report	30
JCLLIB Cross-reference Report	31
INCLUDE Cross-reference Report	32
DCB Cross-reference Report	33
ERROR CODES	34
User ABEND Codes	34
User Return Codes	35

PRODUCT OVERVIEW

Introduction

JCL/ANALYZER is a fast, easy-to-use, cross-reference utility that produces nine reports displaying Job, JCLLIB, Include, Procedure, File and Program relationships. JCL/ANALYZER provides a quick way of identifying every JCLLIB, Include, Procedure, Program and Dataset used in all or any of your Jobs, or all Procedures and Programs that use each of your Files.

JCL/ANALYZER will run on any system that uses Partitioned Datasets (PDSs) for the storage of Procedures (PROCs), JCL, Includes and IDCAMS control statements. Panvalet and Librarian users may use simple Panvalet and Librarian utilities to move Job JCL into PDSs for use with JCL/ANALYZER.

JCL/ANALYZER provides an excellent way to install 'data dictionary' control over Jobs, Procedures, Programs and Datasets. JCL/ANALYZER also gives users an easy way to uniformly and accurately document production job streams. With full Job, Procedure, Program, Dataset and selectivity features, JCL/ANALYZER can be utilized as a powerful problem definition and search utility.

JCL/ANALYZER produces the following:

- A Job Report showing every selected Procedure, Program, Include and File referenced in each Job selected for inclusion in the report.
- A Procedure Report showing every selected Job in the cross-reference run and includes each Program, Include and File referenced within each Procedure.
- A File Report showing every selected Job, Procedure, Include and Program that uses each File identified in the cross-reference run.
- A Program Report showing every selected Job, Procedure and Include referencing each selected Program and each selected File used by each selected Program included in the run.
- A Procedure List which is an abbreviated Procedure Report omitting the file information.
- A PROGRAM List which is abbreviated Program Report omitting the file information.
- A JCLLIB Report showing Job usage, search order and source library name.
- A Include Report showing Job, Procedure and Include usage, level number and source Library names.
- A DCB Report showing all references to RECFM, LRECL and BLKSIZE.
- A file containing a list of all program-names encountered in the JCL.
- A database of all cross-reference information. Can be used for data searches and to produce custom reports.
- In addition to the nine reports produced by JCL/ANALYZER, users may utilize the DASD output produced by JCL/ANALYZER to produce formatted documentation of their own design and specifications.

Features and Capabilities

JCL/ANALYZER has the following features and capabilities:

- JCL/ANALYZER is parameter driven, and requires only simple setup and coding.
- Jobs, Procedures, Datasets or Programs can be selected specifically or generically, or, entire JCL Libraries can be cross-referenced.
- JCLLIBS are searched, in their specified order for Procedures and Includes. Then the PROC datasets are searched.
- INCLUDES used in Jobs and Procedures are read into the cross-reference.
- Instream procedures are treated the same as cataloged procedures.
- Specific or generic Job, Procedure, DD, Dataset or Program names can be included or excluded from processing.
- Reports show related Jobnames, JCLLIBS, Includes, Jobsteps, Procedures, Procsteps, Programs, DDnames, Filenames, and Dispositions for each Dataset identified in the run.
- Symbolic parameters are replaced with their current values, and are underlined on the Job Name Report and flagged as changed on all reports.
- Parameters can be input through the PARM on the EXEC card and/or through the SYSIN DD Statement. The PARM is read first.
- The Dataset name of the PDS from which each Job and Procedure is read is printed behind the Job and Procedure Name on the Job Name Report.
- JCL/ANALYZER dynamically allocates and reads all IDCAMS SYSIN specified Partitioned Datasets to see if any Datasets are dynamically referenced in the Jobstep. Files using the SUBSYS= feature, such as Librarian, will also be allocated.
- Temporary JCL overrides in the Job File may be included in all reports, if requested. The reports will also indicate whether the override is a change or an addition.
- Comment cards may be optionally printed on the Job Name Report.
- Reports may be requested separately or together.
- All JCL/ANALYZER functions can operate with Include or Exclude options such as JOBS, SKIPJOBS, FILES, SKIPFILES, etc.
- IMS and DB2 Batch program names will be replaced with the actual program names in the Parm and/or the Parameter files.
- A Data File containing all of the cross-reference information may be created for use in producing user written reports.
- The product is written completely in Assembler language for maximum efficiency.

INSTALLATION

JCL/ANALYZER is shipped on a labeled tape or cartridge containing one dataset created by IEBCOPY containing the load module MBXREF.

Load Distribution Tape

Create the following JCL which will load the distribution tape to your Load Library

```
//JOBID      JOB (accounting),
//           CLASS=A,MSGCLASS=*,MSGLEVEL=(1,1)
//IEBCOPY EXEC PGM=IEBCOPY
//SYSPRINT  DD SYSOUT=*
//TAPEF1    DD DSN=CSIJAZ.LOADLIB,UNIT=TAPE,DISP=OLD,
//           LABEL=(1,SL),VOL=(,RETAIN,SER=CSIJAZ)
//LOADLIB   DD DSN=your.loadlib,DISP=SHR
//SYSIN     DD *
            COPY I=((TAPEF1,R),O=LOADLIB
/*
//
```

Option Parameters Default Modifications

Some product option parameters can be modified by 'zaps' such that the documented default is reversed. Note that doing this will make the Option Parameters section of this manual incorrect, and will also effectively require repeating this step as a part of future release installations.

Sample Job Control to update a JCL/ANALYZER option default value by zap follows:

```

//JOBNAME JOB ...
// EXEC PGM=AMASPZAP
//SYSLIB DD DSN=loadlibname,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
NAME MBXREF MBXREF
VER offset hexvalue
REP offset hexvalue
/*
    
```

'loadlibname' refers to the PDS library containing the MBXREF load module.

The VER and REP offsets (always the same) and the hexvalues are shown in the table below. See the Option Parameters section for explanation of the individual parameter functions.

Parameter	Normal Default	offset	VER hexvalue	REP hexvalue
UNDERLINES	Yes	0054	E8	D5
OVERRIDES	No	0055	D5	E8
WANT#DB	Yes	0056	E8	D5
IMS	Yes	0057	E8	D5
DB2	Yes	0058	E8	D5
JOBNAMES	No	0059	D5	E8
MEMLIST	Yes	005A	E8	D5
COMMENTS	No	005B	D5	E8
UPDTJOB	No	005C	D5	E8
UPDTPROC	No	005D	D5	E8
LITPOOL	400K	0072	F4F0F0	FxFxFx

'x' in REP value for LITPOOL is any decimal digit, so that FxFxFx is a hexadecimal value for a character number, e.g., F8F5F0 would change the default value to 850 (K). The E8 values are the character 'Y' in hexadecimal, and the D5 values are the character 'N'.

HOW JCL/ANALYZER WORKS

Supported Environments

JCL/ANALYZER will run on any IBM System 9370, S/370, 303x, 308x, 43xx, 3090, ES/9000 or equivalent plug compatible processor, executing under any MVS operating system.

System Requirements

JCL Libraries to be cross-referenced must be in PDSs. JCL statements must be 80 characters in length.

Functional Flow

JCL/ANALYZER is divided into four phases: input scanning, PDS selection, sort invocation and reporting.

1. The input scanning phase edits and builds the users control statements for later use. ABEND code 100 can be returned from this phase if the input control statements are found to be in error.
2. The PDS selection phase includes and/or excludes the specific and/or generic Job name(s), Dataset name(s), DDname(s) and Program name(s). This phase also includes VSAM files dynamically allocated by IDCAMS, retrieves INCLUDE members, resolves symbolics, and includes or excludes temporary JCL overrides. Extracted JCL is written to the SORTIN file for use by the sort program.
3. The sort invocation phase invokes the sort program to sort the extracted data into one or more requested sequences. The JOB report and DATAFILE are not sorted.
4. The reporting phase prints the selected reports.

Member Retrieval

1. JOB members are always retrieved from the file(s) pointed to by the JOB DDname. Procedures found in these file(s) create a Job cross-reference starting at the PROC level.
2. PROCEDURES and INCLUDES are retrieved as follows:
 - a. JCLLIBS in each JOB, if any, are searched first, in their specified order, unless parameter SKIPJCLLIBS=* is coded.
 - b. The file(s) pointed to by the PROC DDNAME are then searched.
3. To increase member processing efficiency, the directories of the JOB and PROC DDNAME files are read into memory. Changes to members in these files, while JCL/ANALYZER is running, will not be reflected on the reports. If concurrent member updates are necessary, the parameters UPDTJOB=Y and UPDTPROC=Y can be used. This will cause the directory to be searched each time a member is processed, which is slower.
4. A compress of the JOB or PROC files during a run may cause a U200 abend.
5. JCLLIBS are allocated as SHR. To reduce overhead, up to twenty JCLLIB files are kept open during a run.

6. IDCAMS.SYSIN parameter files are allocated as SHR and are searched for the names of dynamically allocated files, unless parameter SKIPIDCAMS=* is coded. Up to ten parameter files are kept open during a run.
7. DSNMTV01.DDITV02 and IKJEFTxx.SYSTSIN parameter files are allocated as SHR and are searched for program name replacements, unless IMS=NO or DB2=NO, respectively, are coded. Up to ten parameter files are kept open during a run.

CONTROL STATEMENTS

JCL/ANALYZER is controlled by a series of Control Statements that tell it what functions to perform and what information to include in the PDS search.

There are three types of Control Statements:

1. Function Keywords

The Function Keywords direct JCL/ANALYZER to produce a cross-reference report(s) and/or a file containing the cross-reference data. One or more of the Function Keywords can be specified. If specified, the JOB, PROCEDURE, FILE, PROGRAM, INCLUDE, JCLLIB, PROCLIST, PGMLIST, DCB, PGMFILE and DATAFILE keywords must be specified **before** any Selection Parameters. If no Function Keywords are specified the system will produce a Job Name Report automatically.

2. Option Parameters

The Option Parameters allow the user to alter the report characteristics.

3. Selection Parameters

JCL/ANALYZER processes both Include and Exclude options. Include Parameters select information for inclusion in the cross-reference reports. Likewise the Exclude parameters select information for exclusion from the cross-reference reports.

Control Statement Coding Rules

JCL/ANALYZER control statements are easy to use without always having to refer to the JCL/ANALYZER documentation to determine the syntax rules.

Multiple blanks and commas are ignored;

An equal sign indicates that a sub-parameter follows;

Parameters without subparameters are ignored. ie: SKIPFILES=

An * in column 1 causes the entire line to be ignored.

A one through eight character numeric sequence number that is right justified on the right end of the record and preceded by a space is ignored;

Control Statements may be input through the PARM on the EXEC statement and/or the SYSIN dataset. The PARM is read first.

All control statements are optional.

Function Keywords

The Function Keywords direct JCL/ANALYZER to produce a cross-reference report. One or more of the Function Keywords can be specified. If specified, the JOB, PROCEDURE, FILE, PROGRAM, PROCLIST, PGMLIST, JCLLIB, INCLUDE and DATAFILE keywords must be specified **before** the Selection Parameters.

```
[JOB]      Produce Job Report
[PROCEDURE] Produce Procedure Report
[PROCLIST] Produce Procedure Report without file data
[FILE]     Produce File Report
[PROGRAM]  Produce Program Report
[PGMLIST]  Produce Program Report without file data
[JCLLIB]   Produce JCLLIB report
[INCLUDE]  Produce INCLUDE report
[DCB]      Produce DCB report
[PGMFILE]  Produce file of program names
[DATAFILE] Produce Data File
```

JOB

The JOB keyword requests a Job Report including related JCLLIBS, Procedures, Includes, Programs and Files. Resolved symbolics are underlined. Either JOB or J may be used. JOB is the default and will produce a Job Report unless DATAFILE is encountered in which case only a Data File will be produced. A Job Report starting at the procedure level may be produced by pointing the JOB DD statement to a procedure library.

PROCEDURE

The PROCEDURE keyword requests a Procedure cross-reference Report including related Jobs, Includes, Programs and Files. Either PROCEDURE or C may be used. Note that a JOB report starting at the procedure level may be created if desired. See JOB keyword above.

PROCLIST

The PROCLIST Keyword requests an abbreviated Procedure Report without any file information. The name of the Source Library, where the procedure was found, is included.

FILE

The FILE keyword requests a File cross-reference Report including related Jobs, Procedures, Includes and Programs. Either FILE or F may be used.

PROGRAM

The PROGRAM keyword requests a Program cross-reference Report including related Jobs, Procedures, Includes and Files. Either PROGRAM or P may be used.

PGMLIST

The PGMLIST keyword requests an abbreviated Program Report without any file information. The name of the Source Library, where the program name was found, is included.

INCLUDE

The INCLUDE Keyword requests an Include Report including Job, Procedure, Include level, next higher level Include and Source Library name.

JCLLIB

The JCLLIB Keyword requests a JCLLIB Report including Read-Order, Job and Source Library name.

DCB

The DCB Keyword requests a DCB report showing all uses of RECFM, LRECL and BLKSIZE. Files dynamically allocated by IDCAMS are included.

PGMFILE

The PGMFILE Key word created a file of all program-names encountered in the JCL. Duplicate names are dropped. This file can be input to search or copy programs. The PGMFILE DD statement is required.

DATAFILE

The DATAFILE keyword creates a DATABASE containing all of the cross-reference information. (See the record layout under DATA JCL statement in the Job Control Language section). Reports may also be produced when using this parameter. The DATA DD statement is required.

Note: All reports except the JOB report, and the DATAFILE, are sorted before they are printed.

Option Parameters

The Option Parameters allow the user to alter the reporting characteristics.

The following shows the structure and syntax of the Option Statements.

[PAGE=nnnnn]	Set Lines Per Page
[JOBNAME[S]=JCL]	Use JCL's Jobname
[OVERRIDES=xxx]	Include/Exclude JCL Overrides
[COMMENTS=xxx]	Include/Exclude Comment Cards
[MEMLIST=xxx]	Suppress Member Select List
[DATABASE]=xxx]	Include Database Files
[LITPOOL]=nnn[K]	Dynamically Expand Literal Pool
[UNDERLINES]=xxx]	Underline Resolved Symbolics
[IMS=xxx]	Replace program name DFSRRC00
[DB2=xxx]	Replace program name IKJEFTxx
[IDCABEND=xxx]	Control missing IDCAMS statements
[PROCABEND=xxx]	Control missing PROCEDURES
[INCLUDEABEND=xxx]	Control missing INCLUDES
[JCLLIBABEND=xxx]	Control missing JCLLIBS
[UPDTJOB=xxx]	Allow JOB file updates
[UPDTPROC=xxx]	Allow PROC file updates

Note: Some defaults may be changed during product installation process.

PAGE= nnnn

Sets the lines per page for the cross-reference report(s). The number of lines per page may be set to a value of 1 through 99999. The default lines per page is sixty (60).

JOBNAME[S]= JCL

Use the Job name specified on the JOB statement in the execution JCL instead of setting the Job name equal to the Member name. The default is to use the Member name.

If you request this option, JCL/ANALYZER must look inside every Job to determine the Job name. Because of this, all Member names will appear on the Member Selection List. The reports however, will contain only those members which have been selected by the various Selection Parameters.

OVERRIDES=Y[ES] | N[O]

Include the temporary JCL overrides from the Job File in the cross-reference. The character 'O' will be printed on the right end of the heading, and a 'C' or 'A' will print in the last column to the right of any line that has been overridden. The character 'C' will be printed if the override is a Change. The character 'A' will be printed if the override is an Addition. Since overrides are generally considered to be temporary, the default is OVERRIDES=NO.

COMMENTS=Y[ES] | N[O]

Include the comment cards in the Job Name Report. All comments within selected Jobs, Procedures and Steps will be included. Comments are not included in the Data File when DATAFILE is requested. The default is COMMENTS=NO.

MEMLIST=Y[ES] | N[O]

Suppress the list of the Members selected for inclusion in the cross-reference. The default is MEMLIST=YES which causes the Member List to be produced.

DATABASE=Y[ES] | N[O]

Treat specially coded comment cards as DD statements and include them in the cross-reference process. This option is generally used to include references to database files. The format of the DBMS comment card is:

```
//*XXX [DD] [DSN=]dsname [DISP=]disposition
```

The following explains the contents of the Comment Card shown in the preceding figure:

1. The "/*" is required to indicate that this is a comment.
2. The characters "XXX" represent a three character DBMS-ID beginning in column 4, that must match one of the four ID's patched into the program at install time. This will become the DDname used in the cross-reference. The actual contents of these three characters may vary and are established by the user to his own specifications at the time the product is installed.
3. The dsname is the 1-44 character database file name.
4. The disposition indicates the disposition of the file. It can be the same as any standard JCL disposition or can be INPUT or UPDATE to better define the usage of the file.
5. Parameters must be separated by one or more spaces or commas, must end before column 72 and cannot be continued to a second card.
6. DD, DSN= and DISP= are optional.
7. Cannot be coded within a continued JCL statement.
8. Will be included in the same step that was started by the previous 'EXEC PGM=' statement.
9. This option takes precedence over the COMMENTS=YES option for matching statements.
10. The default is DATABASE=YES unless the user has not patched the program with at least one DBMS-ID at install time.

Example of Use:

```
//*DB2 DD DSN=PAYROLL,DISP=(OLD,KEEP)
//*ADA DSN=GL-MASTER DISP=UPDATE
```

LITPOOL=nnn[K]

Temporarily increase the size of the Literal Pool, for extremely large runs, to the size indicated by "nnn", e.g. LITPOOL=800 would change the size of the Literal Pool to 800K bytes. The default is LITPOOL=400 which directs JCL/ANALYZER to allocate 400K bytes to the Literal Pool. If this parameter is used, it must be the first parameter specified in the PARM. See Error Messages, User ABEND 400 for more information on the literal pool.

UNDERLINES=Y[ES] | N[O]

Suppress the underlining of resolved symbolics on the JOB report. It may be useful to invoke this option when viewing output in the JES queue or when printing a report on a printer that does not support underline mode. The default is UNDERLINE=YES which causes the resolved symbolics

to be underlined. This option may be "hard-wired" to either yes or no when you install the product. See the Installation Guide for the necessary patch.

IMS=Y[ES] | N[O]

Prevents replacing the IMS/DLI program name DFSRRC00 with the second word in the PARM and program name DSNMTV01 with the program name in the parameter file pointed to by DDname DDITV02. If IMS=YES, both program names must pass any SKIPPROGRAM= and PROGRAM= parameters, if entered.

Default = YES

DB2= Y[ES] | N[O]

Prevents replacing DB2 program name IKJEFTxx with the program name in the parameter file pointed to by DDname SYSTIN. If DB2=YES both program names must pass any SKIPPROGRAM= and PROGRAM= parameters, if entered.

Default = YES

IDCABEND=Y[ES] | N[O]

Causes the program to abend with a U200 if an IDCAMS.SYSIN parameter file cannot be allocated or opened. Default is to not abend and to return a U002 at the end of the run. An error message is always written to SYSPRINT.

PROCABEND=Y[ES] | N[O]

Causes the program to abend with a U200 if a Procedure member cannot be found. Default is to not abend and to return a U004 at the end of the run. An error message is always written to SYSPRINT.

INCLUDEABEND=Y[ES] | N[O]

Causes the program to will abend with a U200 if an Include member cannot be found. Default is to not abend and to return a U008 at the end of the run. An error message is always written to SYSPRINT.

JCLLIBABEND=Y[ES] | N[O]

Causes the program to abend with a U200 if a JCLLIB dataset cannot be allocated or opened. Default is to not abend and to return a U012 at the end of the run. An error message is always written to SYSPRINT.

IMSABEND= Y[ES] | N[O]

Causes the program to abend with a U200 if an IMS parameter file pointed to by DDname DDITV02 cannot be allocated or opened. Default is to not abend and to return a U016 at the end of the run. An error message is always written to SYSPRINT.

DB2ABEND= Y[ES] | N[O]

Same as IMSABEND= but applies to DDname SYSTSIN.

UPDTJOB=Y[ES] | N[O]

Allows updates to the members in the files pointed to by the JOB DDname, while JCL/ANALYZER is running. See Note 1 below. Default is UPDTJOB=NO

UPDTPROC=Y[ES] | N[O]

Allows updates to the members in the files pointed to by the PROC DDname while JCL/ANALYZER is running. See Note 1 below. Default is UPDTPROC=NO

Note 1:

Normally, JCL/ANALYZER reads and stores the directory information of the files pointed to by the JOB and PROC DDNAMES. When a Job, Procedure or Include member is needed, this directory information is used for quick member retrieval. This saves run time, but if a member

is changed during the run, the new version of the member is not retrieved.

If it is necessary to update members during a run, use the UPDTJOB=YES and/or the UPDTPROC=YES parameters. This will cause a directory search each time a member is needed, which is slower, but assures retrieval of the latest version of each member. These values can be permanently patched if needed.

JCLLIBS are always searched, in their specified order, before the PROC files, using a directory search. (See the SKIPJCLLIB= parameter).

Selection Parameters

JCL/ANALYZER processes both Include and Exclude options. The Include Parameters are documented in the section, "Include Parameters". Likewise the Exclude Parameters are documented in the section "Exclude Parameters".

JCL/ANALYZER processes the Include parameters first to define the "working set". JCL/ANALYZER then processes the Exclude parameters against the "working set" to define the subset of selection items that are to appear on the report.

Selection Parameter Coding Rules

All selection parameters must be immediately followed by an equal sign ("=").

One or more blank spaces must separate the sub-parameters from each other.

Subparameters can be enclosed in quotes.

Subparameters can be full length or generic(short). To prevent a short subparameter from being used as generic, enclose it in quotes with a space on the right end.

For example, Include and Exclude parameters can specify specific or generic Job names, Dataset names, Program names and DDnames. To select only the program "GL100" and not others that begin with "GL100", the user could code 'GL100 '.

JOB, PROCEDURE, PROGRAM and DDNAME sub-parameters must not be greater than eight characters long.

Any Selection Parameter that is not followed by an Include or Exclude sub-parameter is ignored. This allows parameters to be nullified in the JCL without removing the parameter.

Include Parameters

The Include Parameters are optional. The default is to include **all** Jobs, Procedures, Datasets, Programs and DDnames.

<code>[JOB [S]=]</code>	<code>Include Job (s)</code>
<code>[PROCEDURE [S]=]</code>	<code>Include Procedure (s)</code>
<code>[FILE [S]=]</code>	<code>Include Dataset (s)</code>
<code>[PROGRAM [S]=]</code>	<code>Include Program (s)</code>
<code>[DDNAME [S]=]</code>	<code>Include DDNAME (s)</code>

JOB[S]=

Include only the specific or generic Job names selected on the cross-reference reports.

PROCEDURE[S]=

Include only the specific or generic Procedure names selected on the cross-reference reports.

FILE[S]=

Include only the specific or generic Dataset names selected on the cross-reference report.

PROGRAM[S]=

Include only the specific or generic Program names selected on the cross-reference report.

DDNAME[S]=

Include only the specific or generic DDnames selected on the cross-reference report.

Exclude Parameters

The Exclude Parameters are optional. The default is to not exclude **any** Jobs, Datasets, Programs and DDnames.

[SKIPJOB[S]=]	Exclude Job(s)
[SKIPPROCEDURE[S]=]	Exclude Procedure(s)
[SKIPFILE[S]=]	Exclude Dataset(s)
[SKIPPROGRAM[S]=]	Exclude Program(s)
[SKIPDDNAME[S]=]	Exclude DDNAME(s)
[SKIPIDCAMS[S]=]	Skip Allocating IDCAMS input
[SKIPJCLLIB[S]=]	Skip Allocating JCLLIB(s)

SKIPJOB[S]=

Exclude the specific or generic Job names from the cross-reference reports.

SKIPPROCEDURE[S]=

Exclude the specific or generic Procedure names from the cross-reference reports.

SKIPFILE[S]=

Exclude the specific or generic Dataset names from the cross-reference reports.

SKIPPROGRAM[S]=

Exclude the specific or generic Program names from the cross-reference reports.

SKIPDDNAME[S]=

Exclude the specific or generic DDnames from the cross-reference reports.

SKIPIDCAM[S]=

Suppress the allocation and reading of selected IDCAMS SYSIN datasets to search for dynamically allocated filenames. The option works independently of any SKIPFILES parameters. The option is useful if the user wishes to suppress access to a file to which access is restricted by a security package such as RACF or ACF2. The default is to allocate and read all datasets identified in IDCAMS SYSIN statements. The format of the statement is: SKIPIDCAMS= followed by any number of 1-44 character, full or generic dataset names. SKIPIDCAMS=* will skip all IDCAMS SYSIN files. Does not affect SYSIN DD * files.

SKIPJCLLIB[s]=

Suppress the allocation and reading of selected JCLLIB files for Procedures and Includes. By default, when a Procedure or Include member is referenced, the JCLLIB files for the Job are searched, in their specified order, before the files pointed to by DDname PROC. Use this option to bypass searching one or more JCLLIB files. The format is SKIPJCLLIBS= followed by any number of 1-44 character full or generic dataset names. SKIPJCLLIBS=* will skip all JCLLIB files.

EXAMPLES OF USE

Selection Parameter Examples

The following examples show the syntax of the selection parameters.

```
JOBS= AR100 AP125  
JOBS = AR100 AP125
```

The parameters shown in the first line in the example above are valid, but the second line would produce a syntax error and cause JCL/ANALYZER to terminate with an ABEND code of 100.

```
JOBS=XX100 AA125 SKIPJOBS=XX100Q XX100Z 'AA125RR'
```

In the above example, only Job names that begin with "XX100" or "AA125" are eligible for processing. Out of those Job names, the Job names that begin with "XX100Q" or "XX100Z" are excluded from processing as is the Job name "AA125RR".

```
PROGRAMS=AMS100 SKIPPROGRAMS=AM
```

The parameters shown in this example would not produce the desired results because the PROGRAMS keyword is processed first and it defines the working set. JCL/ANALYZER then applies the SKIPPROGRAMS keyword sub-parameters against the working set to produce the subset. All Programs would be skipped.

```
PROGRAMS=AMS100 SKIPDDNAMES='SORTWK' SYSUDUMP SYSABEND
```

In the above example, the user has defined the working set as all Program names that begin with "AMS100". Out of that working set, the user is excluding the generic DDname "SORTWK" and the specific names of "SYSUDUMP" and "SYSABEND".

JOB CONTROL LANGUAGE

The JCL statements needed to execute JCL/ANALYZER are described in this section.

Job Control Statements

EXEC JCL Statement

The EXEC JCL statement has two functions in JCL/ANALYZER. They are: to execute the JCL/ANALYZER Program; and to optionally input control statements through the PARM field.

The first function is a requirement, but the second function can also be performed by a SYSIN JCL statement. Please note that the operating system allows a maximum of 100 characters in the PARM field. This should not be a problem for most executions of JCL/ANALYZER, but if it is, then use the SYSIN DD statement.

If both the PARM and the SYSIN are used, the PARM is read first.

Sample Execute Statement:

```
//STEP EXEC PGM=MBXREF,PARM='JOB FILE JCLLIB PROCLIST'
```

JOB and PROC JCL Statements

The JOB and PROC DD statements define the Job Libraries and Procedure/Include Libraries, respectively. The JOB and PROC DD statements must point to Partitioned Datasets (PDS). Both the JOB and PROC DD statements can be concatenated and must have a logical record length not greater than eighty (80) characters. The PROC files are searched for Procedures and Includes after any JCLLIBS are searched, in each Job. When members with duplicate names are encountered, the first one is used.

Sample JOB and PROC Statements:

```
//JOB DD DSN=[PROD.JOBLIB,DISP=SHR
// DD DSN=TEST.JOBLIB,DISP=SHR
//PROC DD DSN=PROD.PROCLIB,DISP=SHR
// DD DSN=PROD.INCLLIB,DISP=SHR
```

SYSPRINT JCL Statement

The SYSPRINT statement receives the control statements entered from the SYSIN Dataset and/or the PARM field on the EXEC statement, error messages, and a list of all Jobs extracted from the Job File.

The SYSPRINT statement must specify SYSOUT or a Dataset organization of physical sequential (DSORG=PS) with a record length of 133 bytes (LRECL=133), and BLKSIZE of any multiple of 133. The record format is fixed with ANSI control characters (RECFM=FBA).

REPORT JCL Statement

The REPORT statement receives the report output.

The REPORT statement must specify SYSOUT or a Dataset organization of physical sequential (DSORG=PS) with a record length of 133 bytes (LRECL=133), and BLKSIZE of any multiple of 133. The record format is fixed with ANSI control characters (RECFM=FBA).

SYSOUT JCL Statement

The SYSOUT statement receives the output from the sort program. The user can optionally code this statement as "DD DUMMY". If a SORT problem occurs, assign this DD to SYSOUT. If the installation installed the sort program and specified a DDNAME other than "SYSOUT", this DDNAME must be specified instead.

Sample SYSPRINT, REPORT, and SYSOUT Statements:

```
//SYSPRINT DD   SYSOUT=*  
//REPORT  DD   SYSOUT=*  
//SYSOUT  DD   DUMMY
```

SORTIN JCL Statement

The SORTIN JCL statement must be specified and is used by the installation's sort control program. The sort program is dynamically invoked by JCL/ANALYZER. Records 140 bytes long and blocked at 8680 are written to the SORTIN file before the sort is invoked. There is approximately one record for each selected DD statement. DCB parameters are provided by JCL/ANALYZER. Blocksize can be overridden in the JCL. For faster access, this file can be assigned to VIO.

SORTWKnn JCL Statements

The SORTWKnn JCL statements must be specified and are used by the sort control program. The sort program is dynamically invoked by JCL/ANALYZER. SORTWK01 through SORTWK03 are required. Additional SORTWKnn DD statements can be specified to enhance the speed of the sort processing (See Sort manual). Space requirements are dependent on the amount of data being cross-referenced.

Sample SORTIN and SORTWKnn Statements:

```
//SORTIN   DD SPACE=(CYL,(10)),UNIT=SYSDA  
//SORTWK01 DD SPACE=(CYL,(10)),UNIT=SYSDA  
//SORTWK02 DD SPACE=(CYL,(10)),UNIT=SYSDA  
//SORTWK03 DD SPACE=(CYL,(10)),UNIT=SYSDA
```

PGMFILE JCL Statement

The PGMFILE JCL statement defines a file to receive a list of all selected program-names, when the PGMFILE function is requested. There is one record per program-name. Duplicates are dropped. The default DCB=s are RECFM=FB, LRECL=80, and BLKSIZE=800. The BLKSIZE can be overridden on the JCL.

Sample PGMFILE statement:

```
//PGMFILE DD   DSN=PGM.FILE,DISP=(New,CATLG),  
//          UNIT=SYSDA,SPACE=(CYL,(1,1))
```

DATA JCL Statement

The DATA JCL statement defines a file to hold a DATABASE of all of the selected JCL data, when the DATAFILE function is requested. The DATABASE can be used for searches or to produce custom reports.

DATA#REC	DISP	PIC	LRECL=332)
MEM#NM	001	CL8	MEMBER NAME (Used as JOBNAME unless JOBNAME=JCL)
CNTR1	009	ZL8	SEQUENTIAL COUNT from beginning of run
JCL#NM	017	CL8	JCL NAME (JOBNAME from job card unless JOBNAME=JCL)
MEM#SRC	025	CL44	SOURCE, PDS the JOB member came from
INCL#NM	069	CL8	INCLUDE NAME
INCL#LVL	077	CL2	INCLUDE LEVEL, 01-16 00=NO INCLUDE
INCL#SYM	079	CL1	INCL HAD SYMBOLICS from <u>Exec</u> , <u>Proc</u> , <u>Set</u> or <u>Mult</u> .
INCL#SRC	080	CL44	INCLUDE SOURCE (PDS INCLUDE was found in)
EXE#STP	124	CL8	EXEC STEP- EXEC PGM(JOB) EXEC PROC(JOB/PROC)
PROC#NM	132	CL8	PROCEDURE NAME
PROC#LVL	140	CL2	PROC LEVEL, 01-16 00=NO PROC
PROC#SYM	142	CL1	PROC HAD SYMBOLICS from <u>Exec</u> , <u>Proc</u> , <u>Set</u> or <u>Mult</u> .
PROC#SRC	143	CL44	PROC SOURCE (PDS PROC was found in)
STEPNO	187	ZL4	STEP NUMBER within this JOB from EXEC PGM=
PROC#STP	191	CL8	PROC STEP NAME from EXEC PGM= in PROC
PGM#NM	199	CL28	PROGRAM NAME (LG=28 for backward reference)
PGM#SYM	227	CL1	PGM HAD SYMBOLICS from <u>Exec</u> , <u>Proc</u> , <u>Set</u> or <u>Mult</u> .
DDNAME	228	CL8	DDNAME
FILENAME	236	CL48	FILE NAME (May include member)
UNITFLG	284	CL1	'A' if UNIT=AFF, else= space
UNIT	285	CL8	DEV-NO,DEV-TYPE,GRP or DDNAME if UNITFLG=A
RECFM	293	CL3	RECORD FORMAT
LRECL	296	CL5	LOGICAL RECORD LENGTH
BLKSIZE	301	CL5	BLOCKSIZE
FILE#SYM	306	CL1	FILE HAD SYMBOLICS from <u>Exec</u> , <u>Proc</u> , <u>Set</u> or <u>Mult</u> .
DISP	307	CL3	DISP=S,C,O,N,D,P,M,U,K,I(INPUT), X(UPDATE)
DISP#SYM	310	CL1	DISP HAD SYMBOLICS from <u>Exec</u> , <u>Proc</u> , <u>Set</u> or <u>Mult</u> .
OVREFLG	311	CL1	C=OVERRIDE CHANGE, A=OVERRIDE ADD
DYNAM	312	CL1	DYNAM ALLOC: A=BAD ALLOC, O=BAD OPEN, S=SUBSYS NOT FOUND, MEMBER NOT FOUND, F=FILE FROM DYN ALLOC READ.
NM#SW	313	CL1	Y=MEM#NM and JCL#NM switched (JOBNAME=JCL)
FILLER	314	CL19	FILLER

The information contained in the Database is formatted as follows:

The DATA file defaults to: RECFM=FB, LRECL=332 and BLKSIZE=8300. Blocksize can be overridden in the JCL. There is approximately one record for each selected DD statement.

Sample DATA Statement:

```
//DATA DD DSN=X.Y.Z,DISP=(NEW,CATLG),UNIT=SYSDA,
//      SPACE=(CYL,20,10))
```

SYSIN JCL Statement

The SYSIN DD statement is used to provide control statements to JCL/ANALYZER. The record format must be either fixed unblocked or fixed blocked, and the Dataset organization must be either physical sequential or partitioned. If a member of a PDS is used as input to JCL/ANALYZER, then the member name must be coded on the SYSIN DD statement.

The PARM field of the EXEC statement is first examined to see if it was coded. If the PARM field was coded, the control statements are used by JCL/ANALYZER. Next, JCL/ANALYZER examines an internal operating system control block to determine if the SYSIN DD statement was coded. If the SYSIN DD statement was coded, then JCL/ANALYZER opens the SYSIN dataset and reads JCL/ANALYZER control statements.

If neither the PARM field of the EXEC statement nor the SYSIN DD statement are specified, JCL/ANALYZER will produce a Job Name Report by default using the entire JOB and PROC Libraries as specified in the JOB and PROC DD statements.

Sample SYSIN Statements:

```
//SYSIN DD DSN=MBXREF.CNTL(MBXREFJF),DISP=SHR  
//SYSIN DD *
```

Sample Job Execution

An example of the JCL needed to execute JCL/ANALYZER and an explanation of each statement follows:

```
//XREF      JOB  (accounting-info),name,CLASS=x
//          EXEC  PGM=MBXREF,PARM='LITPOOL=200K'
//JOB       DD  DSN=PROD.JOBLIB,DISP=SHR
//PROC      DD  DSN=PROD.PROCLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=A
//REPORT    DD  SYSOUT=A
//SYSOUT    DD  DUMMY
//SORTIN    DD  SPACE=(CYL,(10)),UNIT=SYSDA
//SORTWK01 DD  SPACE=(CYL,(10)),UNIT=SYSDA
//SORTWK02 DD  SPACE=(CYL,(10)),UNIT=SYSDA
//SORTWK03 DD  SPACE=(CYL,(10)),UNIT=SYSDA
//SYSIN     DD  *
JOB FILE   PROCLIST
JOBS= AR AP
SKIPFILES=DUMMY NULLFILE
/*
```

The following explains each of the statements shown in the sample JCL shown above:

1. The JCL/ANALYZER control program is executed with a PARM statement specifying that the Literal Pool is to be set to 200K bytes. A PARM field could have been used to contain the remainder of the control statements as well since the control statements used less than 100 bytes.
2. The JOB DD statement defines the PDS that contains the Jobs to be cross-referenced.
3. The PROC DD statement defines the PDS that contains the Procedures and Includes to be cross-referenced.
4. The JCL/ANALYZER control statements are sent to the SYSPRINT DD statement.
5. The cross-reference reports are sent to the REPORT DD statement.
6. The sort messages are sent to the SYSOUT DD statement.
7. SORTIN and SORTWK01 through SORTWK03 are directed to the SYSDA unit.
8. The SYSIN DD statement defines the input to JCL/ANALYZER.
9. The JOB, FILE and PROCLIST Reports are requested. JCL/ANALYZER is directed to select only those Job Names beginning with "AR" or "AP", and to skip all files that are specified as "DD DUMMY" or "DSN=NULLFILE".

FILE Cross-reference Report

-----CROSS REFERENCE BY FILE										MARCH 19, 1994 13.41 PAGE 001		
FILENAME	JOBNAME	JOBSTEP	PROCEDURE	PROCSTEP	PROGRAM	DDNAME	DISPOSITION	INCLUDE	FCPDI	O		
GL.CONTROLS	GLD0200	PROC02	GLD0211	STEP001	IDCAMS	INDD1	SHR			A		
	GLD0500	STEP001	GLD0500	IDCAM001	IDCAMS	SYSIN	OLD DEL	KEP				
				IDCAM001	IDCAMS	SYSIN	NEW CAT	DEL				
	MBD109	MBSTEP01	MBD109	STEP35	MBX123	DD20	SHR	GL2016				
GL.CONTROLS.BACKUP(+1)	GLD0200	PROC02	GLD0211	STEP001	IDCAMS	OUTDD1	NEW CAT	DEL	E			
				ABEND015	IEFBR14	DD1	OLD DEL	DEL				
GL.CONTROLS.DATA	GL0500	STEP001	GLD0500	IDCAM001	IDCAMS	SYSIN	NEW CAT	DEL		C		
GL.CONTROLS.INDEX	GLD0500	STEP001	GLD0500	IDCAM001	IDCAMS	SYSIN	NEW CAT	DEL				
GL.DAILY	GLD0200	PROC01	GLD0105	BKUP050	IDCAMS	GLX0001	SHR					
				RPT100	GL5102	FILE07	SHR	KEP	KEP			
				UPDT200	GL5105	GLFILE1	SHR					
		PROC03	GLD0105	BKUP050	IDCAMS	GLX0001	SHR					
	RPT100			GL5102	FILE07	SHR	KEP	KEP				
	UPDT200			GL5105	GLFILE1	SHR						
	GLD0500	STEP001	GLD0500	GLSTEP5	GL4110	GL0043	SHR		P			
GL.DAILY.BACKUP(+1)	GLD0200	PROC01	GLD0105	BKUP050	IDCAMS	GLX0002	NEW CAT		E	E		
		PROC03	GLD0105	BKUP050	IDCAMS	GLX0002	NEW CAT					
GL.MASTER	GLD0200	PROC02	GLD0211	STEP001	IDCAMS	INDD2	SHR					
GL.MASTER.BACKUP(+1)	GLD0200	PROC02	GLD0211	STEP001	IDCAMS	OUTDD2	NEW CAT	DEL				
				ABEND015	IEFBR14	DD2	OLD DEL	DEL	SE	E		

The example above shows the following:

1. Filename GL.CONTROLS is used in three procedures, three jobs and one include.
2. The P, E, S and M under heading FCPDI specify whether a SYMBOLIC in a File, proC, Program, Disposition or Include was replaced by a value in a Proc, Exec or Set statement or Multiply values.
3. The A and C under heading O shows overrides that are Adds or Changes.

JCLLIB Cross-reference Report

-----CROSS REFERENCE LIST OF JCLLIBS				JULY 06, 1994	15.16	PAGE 001
JCLLIB	ORDER	JOBNAME	SOURCE			
PROD.PROCLIB1	01	JOBA	PROD.JOBLIB			
	01	JOBB	PROD.JOBLIB			
PROD.PROCLIB2	02	JOBA	PROD.JOBLIB			
	02	JOBB	PROD.JOBLIB			
PROD.PROCLIB3	03	JOBA	PROD.JOBLIB			
	03	JOBB	PROD.JOBLIB			

The example above shows the following:

1. PROD.PROCLIB1, PRO.PROCLIB2 and PROD.PROCLIB3 are all used in JOBA and JOBB.
2. The number under the heading 'ORDER' shows each JCLLIB'S search order within each job.
3. The source library where each job was found is shown.

INCLUDE Cross-reference Report

-----CROSS REFERENCE LIST OF INCLUDES								JULY 06, 1994	15.16	PAGE 001
INCLUDE	LVL	SYM	JOBNAME	PROCEDURE	LVL	SYM	W/N-INCL	SOURCE		
GLI002	01	S	JOBA	PROC001	01			PROC.INCLLIB1		
	01		JOBB	PROC003	01	E		PROC.INCLLIB1		
GLI004	02		JOBA	PROC001	01		GLI002	PROC.INCLLIB1		
GLI006	03		JOBA	PROC007	01		GLI004	PROC.INCLLIB2		

The example above shows the following:

1. Include GLI002 is referenced in jobs JOBA and JOBB within procedures PROC001 and PROC003.
2. The Include level shows the nested level of the include within other includes.
3. Include GLI004 is referenced in include GLI002. Include GLI006 is referenced in include GLI004.
4. The 'S' under the first 'SYM' heading shows that the include name GLI002 was altered by a Set statement (Set, Exec or Proc).
5. The 'E' under the second 'SYM' heading shows that the procedure name PROC003 was altered by an Exec statement.
6. The 'SOURCE' heading lists the libraries that the includes were found in.

DCB Cross-reference Report

-----DCB CROSS-REFERENCE REPORT										JULY 06, 1994		15.16		PAGE 001	
FILENAME	RCF	LRECL	BLKSIZE	JOBNAME	PROCEDURE	PROGRAM	DDNAME	DISPOSITON	INCLUDE	FCPDI	O				
&&XX2	FB	72	7200	IEBCOMP	COMPAR05	IEBCOMP	DD4	NEW PAS DEL							
DATAKICK.INPL	VB	9996	10000	MBJOB01	MBPROC04	IEBGENER	SYSUT2	NEW KEE		P					
MSB.DOC.LIB9	FB	80	3120	IEBUPDT		IEBUPDTE	SYSUT2	NEW CAT DEL			A				
MBS.PARMLIB(SCNJCL02)	FB	500	5000	MBJ001	MBP001	SCNCOB01	SYSTSIN	OLD	MBINCL01						
MBS.SPAED	VBS	200	2000	IDCAMS2		IDCAMS	DD1	NEW CAT DEL							
MBTAPE	F	80	80	IEBGENER		IEBGENER	SYSUT1	OLD							

The example above shows the following:

1. All file names that have DCB parameters in the JCL, or in IDCAMS parameters.
2. The specified RECFM, LRECL and BLKSIZE.
3. The related JOB, PROC, PROGRAM, DDNAME and INCLUDE.
4. The DISPOSITION showing how each file is used.
5. The FCPDI fields show if a File, ProC, Program, Disposition or Include was replaced by a SYMBOLIC from a Proc, Exec or Set Statement or by Multiple values.
6. The Override field shows if an override was an Add or a Change.

ERROR CODES**User ABEND Codes**

CODE	EXPLANATION
100	CONTROL STATEMENTS IN ERROR JCL/ANALYZER prints the control statements on the SYSPRINT DD statement and places an asterisk ("*") under the keyword in error.
200	DATASET OPEN OR I/O ERROR JCL/ANALYZER encountered an open error or an I/O error while processing. See the accompanying error message for more details.
300	SORT ERROR The sort program returned an error condition. The sort error message can be found on the SYSOUT DD statement listing.
400	LITERAL POOL OVERFLOW The literal pool size has been exceeded. See your system programmer to permanently increase the size of the literal pool. The installation manual contains instructions explaining how to permanently increase the size of the literal pool. As a temporary measure you may insert LITPOOL=nnn[K] as the first parameter in the PARM on the EXEC card. 'nnn' represents the size of the literal pool in thousands of bytes. The program is normally installed with the literal pool set to 400K. you may increase this size up to 999K. The literal pool contains all of the JCL/ANALYZER parameters as well as a twelve digit entry for each JOB Library member selected to be included in the cross-reference. For example , 5,000 selected JOB members would require that the literal pool be set to at least 60K plus whatever space is required to contain the cross-reference parameters.
410	TOO MANY PROCEDURE/INCLUDE MEMBERS Remove some concatenations from the PROC file(s).
600	JCL/ANALYZER HAS EXPIRED The expiration date has been reached and JCL/ANALYZER terminates processing.

User Return Codes

These codes do not cause an ABEND and always display an informative message on SYSPRINT. At the end of the run, these return codes are added together to create the final user Return Code.

CODE	EXPLANATION
002	IDCAMS.SYSIN PARAMETER FILE COULD NOT BE ALLOCATED OR OPENED See SKIPIDCAMS= selection parameter.
004	PROCEDURE MEMBER COULD NOT BE FOUND See SKIPPROCEDURES= selection parameter
008	INCLUDE MEMBER COULD NOT BE FOUND
012	JCLLIB FILE COULD NOT BE ALLOCATED OR OPENED See SKIPJCLLIBS= selection parameter.
016	IMS/DB2 PARAMETER FILE COULD NOT BE ALLOCATED OR OPENED. See IMS= and DB2= option parameters.

Note: The above errors can be changed to an ABEND. See the IDCAMSABEND=YES, PROCABEND=YES, INCLUDEABEND=YES, JCLLIBABEND=YES, IMSABEND=YES and DB2ABEND=YES option parameters.