Debug Tool

Load Module Analyzer

Stand-Alone User Documentation

Version 6.1.1

Wednesday, November 02, 2005
1:48 PM

Contents

Overview

The Debug Tool Load Module Analyzer (LMA) is a stand-alone program designed to analyze MVS load modules or program objects to determine the language translator (compiler or assembler) used to generate the object for each CSECT in the load module.

Invocation

The Load Module Analyzer is invoked using standard MVS JCL. Member JCLSAMP provided in the DATA library contains JCL that can be used to invoke LMA.

The remainder of this section describes the JCL used to run LMA.

By default the LMA program processes all members in the PDS or PDSE pointed to by the EQALIB DD statement. However, control statements can be used to select only specific members of this dataset.

DD-Names used by the Debug Tool Load Module Analyzer

The following DD-Names are used by this program:

EQALIB – the load modules to be analyzed.

> This may be a concatenation of PDS or PDSE data sets. However, if the same member is present in more than one of the concatenated datasets, only the first will be processed.

EQAPRINT – the output report.

> It can be RECFM=FBA with LRECL >=133 or VBA with LRECL >= 137.

EQAIN – control statements.

> These statement have the form “SELECT MEMBER=xxxx” to select only certain load modules for processing. If all load modules are to be processed, this DD can be omitted, directed to a DUMMY, or directed to an empty dataset.
>
> This file must be RECFM=FB and LRECL=80. Each control statement must be on a separate line. The entries are free-form and blanks may occur before or after each keyword and operator. Comments may be included by placing an asterisk in column 1.

EQASYSPF – a list of “system” prefixes.

> When a CSECT is found that corresponds to one of these prefixes and the entry for the prefix indicates that CSECT name’s beginning with this prefix are not to be included, individual entries are not displayed for these CSECT’s. Instead, a

single line is output indicating that CSECT's with the prefix were found in that load module.

This file must be RECFM=FB and LRECL=80. Data for this file is provided in member EQALMPFX of the table library (SEQATLIB) shipped with Debug Tool. See "EQASYSPF file format" below for directions on how to provide customized entries for this file.

EQAPGMNM – a list of program names corresponding to program ID's found in the load module IDR data.

This file must be RECFM=FB and LRECL=80. Data for this file is provided in member EQALMPGM of the table library (SEQATLIB) shipped with Debug Tool. See "EQAPGMNM file format" below for directions on how to provide customized entries for this file.

Parameters

The parameter string passed to this program may consist of any of the following separated by commas or blanks:

CKVOLFPRS – lists only CSECT's or entries that use the Additional Floating-Point Registers in a volatile manner. This parameter cannot be specified with the OSVSONLY parameter. If both are specified, the last one will be used.

This parameter is not currently supported via the ISPF interface. It can be specified only using JCL.

DATEFMT=*dateformat* – indicates how dates are to be formatted.

Note that if a date from the binder CSECT identification record (IDR) data does not appear to be a valid Julian date, it will not be reformatted. The possible values of *dateformat* are:

YYYYMMDD – sort format YYYY/MM/DD. (This is the default.)
MMDDYYYY – U.S. standard: MM/DD/YYYY
DDMMYYYY – European standard: DD/MM/YYYY

LEINFO – causes the text for each CSECT and external entry point to be inspected for a Language Environment footprint. If one is found information about the Language Environment entry point name, linkage type, source language, and translation date and time is included in the output for the CSECT or entry. If no Language Environment footprint is found, the prologue code is inspected for known non-Language Environment prologue formats. If one is discovered the corresponding language is included in the output. Otherwise, "ASSEMBLER" is output.

This parameter is not currently supported via the ISPF interface. It can be specified only using JCL.

LESCAN – causes the actions described under the LEINFO parameter. In addition, the text for each CSECT is scanned looking for "hidden" Language Environment entry points that do not correspond to an external symbol. These may be present for C static functions, etc. If such "hidden" entry points are detected, the same output as described for LEINFO in generated.

This parameter is not currently supported via the ISPF interface. It can be specified only using JCL.

LISTLD – lists all LD (label definition) entries in addition to CSECT names

LOUD – causes the data read from the EQASYSPF and EQAPGMNM files to be shown in the output listing.

NATLANG=*lll* – indicates the national language to be used.

The possible values of *lll* are:

ENU – mixed-case English. (This is the default.)
UEN – upper-case English.
JPN – Japanese. *******NOT CURRENTLY SUPPORTED*******

OSVSONLY – only list CSECT's compiled with the OS/VS COBOL compiler. Information about all other CSECT's will be suppressed. This parameter cannot be specified with the CKVOLFPRS parameter. If both are specified, the last one will be used.

SHOWLIB – overrides the include indicator for all system library prefixes specified in the EQASYSPF file so that all CSECT's are listed.

SORTBY=*sortopt* – indicates how the output CSECT's are to be sorted.

The possible valudes of *sortopt* are:

OFFSET – sorts by offset (the order shown in the linkage editor or AMBLIST output. (This is the default.)

NAME – sorts by CSECT name

PROGRAM – sorts by the translator program ID

LANGUAGE – sorts by the source language and by the translator program ID.

DATE - sorts by the translation date.

EQASYSPF file format

This file contains a list of "system" prefixes. When a CSECT is found that corresponds to one of these prefixes and the entry for the prefix indicates that CSECT names's beginning with this prefix are not to be included, individual entries are not displayed for these CSECT's. Instead, a single line is output indicating that CSECT's with the prefix were found in that load module.

Data for this file is provided in member EQALMPFX of the table library (SEQATLIB) shipped with Debug Tool. However, you may wish to add additional entries for this file. If so, you should create a data set containing your additional entries and concatenate this data set to the one shipped with Debug Tool. If, in the future, updates are made to the data shipped with Debug Tool, you will then automatically pick up those updates along with your additions.

Each line in this file represents one entry. The entries are free form but each item must be separated from the previous item by one or more blanks. Comments may be included by placing an asterisk in column 1. The format is as follows:

xxx I L *description*

where:

xxx – a one to seven character prefix

I –Include Indicator. '1' indicates that each CSECT beginning with this prefix is to be treated as an ordinary CSECT and '0' indicates that these names are not to be listed individually.

L – Language / System Component Indicator.

Choose from one of the following characters:

B - COBOL
V – OS/VS COBOL
P – PL/I
E – Enterprise PL/I
C – C/C++
A - Assembler
L – Language Environment
S - CICS
I - IMS
2 – DB2
M - MVS

T – TCP/IP
***** - unclassified

description – a twelve-character description of the component owning the prefix

EQAPGMNM file format

This file contains a list of program names corresponding to program ID's found in the load module IDR data. These names are used in the program output to describe the language translator used to generate the object for the corresponding CSECT.

Data for this file is provided in member EQALMPGM of the table library (SEQATLIB) shipped with Debug Tool. However, you may wish to add additional entries for this file. If so, you should create a data set containing your additional entries and concatenate this data set to the one shipped with Debug Tool. If, in the future, updates are made to the data shipped with Debug Tool, you will then automatically pick up those updates along with your additions.

Each line represents one entry. The entries are free form. The program number must begin in column 1 and each item must be separated from the previous item by one or more blanks. Comments may be included by placing an asterisk in column 1. Sequence numbers are not permitted in this file. The format is as follows:

xxxxxxx L *program-description*

where:

xxxxxxx – a seven character program number

L – Language / System Component Indicator. See the description of "Language / System Component Indicator" in the description of the EQASYSPF file for a list of possible values.

program-description – a description of the program.

Program Output

The general ordering of the output for each load module or program object is as follows:

- All members of the first EQALIB concatenation with each load module or program object appearing in alphabetical order
- All members of the second EQALIB concatenation that are not duplicates of members in the previous concatenation with each load module or program object appearing in alphabetical order

- …

However, alias names are handled differently:

- If a member by the "true name" exists, the output for this "true name" appears in the general order previously described. Prior to the output of the contents of that member, a list of alias names for the member is given.
- If the "true name" is not present in the data set, the alias appears as part of the general order previously described.

Output contents

The following information is included in the output for each CSECT:

- CSECT name
- CSECT offset in load module or segment
- CSECT length
- Program-ID as contained in the binder IDR data
- Translator (compile or assembly) date
- Program description as supplied for the specified program ID
- For OS/VS COBOL only, PARM=RES or PARM=NORES
  PARM=RES indicates that all OS/VS COBOL CSECT's in the load module or program object were compiled with the NORES compiler option.
- If LEINFO, LESCAN, or CKVOLFPRS is specified either:
  - If a Language Environment prologue was detected information will be included in a string identified by LEINFO=(…. This string contains the Language Environment entry name or an asterisk to indicate that the name is the same as the external symbol, Language Environment linkage type, source language, and translation date, time, and translator version.
  - If no Language Environment prologue was detected, but the prologue appears to be that of a known, non-LANGUAGE environment compiler, on of the following will be included: C/C++, COBOL, or PL/I.
  - Otheriwse, ASSEMBLER will be included to indicate that the program is likely to be an assembler program.

Sample Output

The following shows an example of the output from this program,

The first page shows any PARM string passed to the program and any SELECT statements read from the EQAIN DD.

```
1


 Parm:
 NATLANG=ENU,DATEFMT=YYYYMMDD,SORTBY=OFFSET

 EQAIN:

  SELECT MEMBER=IPTH13
```

The next two pages are shown only if the LOUD option is specified. They show the system prefixes and program numbers in effect.

```
1 5655-M19   Debug Tool Version 6 Release 1 Load Module Analyzer  04/22/2005  12:35     Page   1
 EQASYSPF statements:

AQA      Masked (PL/I)           PL/I
BPX      Masked (MVS)            USS
CBC      Masked (C/C++)          C/C++
CEE      Masked (LE)             LE
CEL      Masked (LE)             LE
CLB      Masked (C/C++)          C/C++
DFH      Masked (CICS)           CICS
DFS      Masked (IMS)            IMS
DSN      Masked (DB2)            DB2
EDC      Masked (C/C++)          C/C++
EZA      Masked (TCP/IP)         TCP/IP
IBM      Masked (PL/I)           PL/I
IEDC     Masked (C/C++)          C/C++
IEL      Masked (PL/I)           PL/I
IEN      Masked (PL/I)           PL/I
IGY      Masked (COBOL)          COBOL
IGZ      Masked (COBOL)          COBOL
IIBM     Masked (PL/I)           PL/I
ILB      Masked (OS/VS COBOL)    OS/VS COBOL
```

```
1 5655-M19   Debug Tool Version 6 Release 1 Load Module Analyzer  04/22/2005  16:20     Page   3
 EQAPGMNM statements:

0000000 (?)                (NO IDR DATA AVAILABLE)
360SAS0 (Asm)              S/360 OS ASSEMBLER (F)
5645001 (C/C++)            C/C++ FEATURE OF OS/390 R2
5647A01 (C/C++)            C/C++ FEATURE OF OS/390 R4
5648A25 (COBOL)            COBOL FOR OS/390 & VM VERSION 2
5655B22 (PL/I)             VISUALAGE PL/I FOR OS/390 VERSION 2 RELEASE 2
5655G53 (Enterprise PL/I)  ENTERPRISE COBOL FOR z/OS AND OS/390 VERSION 3
5655H31 (Enterprise PL/I)  ENTERPRISE PL/I FOR z/OS AND OS/390 VERSION 3
5655121 (C/C++)            C/C++ FOR MVS/ESA VERSION 3
5655122 (?)                DB2 VERSION 2
5668767 (?)                VS PASCAL
5668899 (?)                APL2 VERSION 1
5668909 (PL/I)             OS PL/I VERSION 2 RELEASE 1,2, & 3
5668910 (PL/I)             OS PL/I VERSION 2 RELEASE 1,2, & 3
5668958 (COBOL)            VS COBOL II VERSION 1 RELEASE 3
5668962 (Asm)              ASSEMBLER H VERSION 1 RELEASE 2, 3, OR 4
5668806 (Fortran)          VS FORTRAN V2
5688023 (COBOL)            VS COBOL II VERSION 1 RELEASE 4
5688040 (C/C++)            C/370 COMPILER V1
5688187 (C/C++)            C/370 COMPILER V2
5688197 (COBOL)            COBOL FOR MVS & VM VERSION 1 RELEASE 2
5688216 (C/C++)            AD/CYCLE C/370
5688228 (?)                APL2 VERSION 2
5688235 (PL/I)             PL/I FOR MVS & VM VERSION 1 RELEASE 1
5694A01 (C/C++)            C/C++ FEATURE OF z/OS R5
5695013 (?)                REXX/370
5695PMB (Binder)           Z/OS BINDER
5696234 (Asm)              HIGH LEVEL ASSEMBLER FOR MVS & VM & VSE VERSION 1 RELEASE 4
5734AS1 (Asm)              OS ASSEMBLER H
5734PL1 (PL/I)             PL/I OPTIMIZING COMPILER
5740CB1 (OS/VS COBOL)      OS/VS COBOL VERSION 1 RELEASE 2.4
5740RG1 (?)                OS/VS RPG II
5741SC1 (Asm)              OS/VS ASSEMBLER (XF)
SAS/C/! (C/C++)            SAS C
SAS/C/  (C/C++)            SAS C
SAS/C   (C/C++)            SAS C
SAS/C!  (C/C++)            SAS C
```

The following page is an example of the output for a single load module.

```
1 5655-P15   Debug Tool Version 6 Release 1 Load Module Analyzer   10/10/2005  07:29    Page   15
           Load Module  TSCODEL.CICS.TEST.LOAD(CICK512)

 CSECT    Sg   Offset  Length  Program-ID  Trn-Date    Program-Description
 $PRIV000010
                  28     C58   5688216    1996/12/31  AD/Cycle C/370
 $PRIV000011
                 D00    1CD0   5688216    1996/12/31  AD/Cycle C/370
 @@XINIT@        29E0       8   5688216    1996/12/31  AD/Cycle C/370
 @@INIT@         29E8     3D8   5688216    1996/12/31  AD/Cycle C/370
 EQADCRXT        2DC0     240   566896201  1995/05/15  Assembler H Version 1 Release 2, 3, OR 4
 @@C2CBL         3118      10   569623400  1995/08/03  High Level Assembler for MVS & VM & VSE Version 1
 @@FETCH         3138      10   569623400  1995/08/03  High Level Assembler for MVS & VM & VSE Version 1
 MEMSET          3148      10   569623400  1995/08/03  High Level Assembler for MVS & VM & VSE Version 1
 FPRINTF         3158      10   569623400  1995/08/03  High Level Assembler for MVS & VM & VSE Version 1
 CS9403          3168    3518   566895807  1995/08/15  VS COBOL II Version 1 Release 3
 STRLEN          7398      10   569623400  1995/08/03  High Level Assembler for MVS & VM & VSE Version 1
 CEE*                          (Multiple program ID's)
 DFH*                          5668962                Assembler H Version 1 Release 2, 3, OR 4
 EDC*                          5696234                High Level Assembler for MVS & VM & VSE Version 1
 IGZ*                          5668962                Assembler H Version 1 Release 2, 3, OR 4
```

Messages

EQALM000S ** UNKNOWN ERROR ******

Explanation: An unexpected or unrecognized error has occurred.

EQALM001E ** Unable to open *filename* ******

Explanation: This indicated file cannot be opened.

EQALM002I ** Unable to open EQAIN. All members will be processed. ******

Explanation: The EQAIN file was not allocated. All members of the PDS(E) concatenation allocated to EQALIB will be processed.

EQALM003E ** Unknown member specified in SELECT statement ******

Explanation: The member specified on the SELECT statement was not found in the EQALIB concatenation.

EQALM004E ** Unrecognized control statement ******

Explanation: An unrecognized control statement was encountered while processing the EQAIN file.

EQALM005E ** Work area overflow ******

Explanation: An internal work area has overflowed.

EQALM006E ** Error *rc-reason* returned from Binder API ******

Explanation: The indicate return and reason codes were returned from the Binder API's used to extract the IDR data from the load module or program object. See the appropriate Binder documentation for a description of this error.

EQALM007W ** *text* is an unrecognized option ******

Explanation: The specified text is not a supported option.

EQALM008S ** Debug Tool Utilities and Advanced Functions is required to use this function. ******

Explanation: A valid license for this program could not be found on the current system.

EQALM009S ** Unable to load EQALMER2. ******
Explanation: The indicated load module could not be found in the current STEPLIB, system link-list, etc. This program is part of the SEQAMOD data set shipped with Debug Tool.

EQALM010E ** Unable to obtain list of EQALIB members. RC *DEServ-RC* returned from DESERV macro. ******
Explanation: The indicated return code was generated by the DEServ function. Refer the the appropriate Data Facility Product publication for a description of the error.

****** End of document ******