

COPE for IMS New Features Bulletin



www.compuware.com

Email: mainframesolutions@compuware.com

Compuware Headquarters:

1 Campus Martius
Detroit, MI 48226 USA
313-227-7300

DELTA for IMS© is a trademark of the BMC Corporation.

Xpeditor© is a trademark of the Compuware corporation.

SmartTest© is a trademark of the ViaSoft corporation.

HourGlass© is a trademark of Mainware Corporation.

TicToc© is a trademark of Isogon Corporation.

IBM is a registered trademark of International Business Machines Incorporated.

All other trademarks and service marks are the property of their respective owners.

Copyright © 1989-2012, Compuware Corp. All rights reserved.

This material may not be reproduced in whole or in part by any means, without written permission from Compuware Corp.

Contents

Contents

Preface.....	6
About This Manual	6
COPE for IMS/DC New Features Bulletin	6
Related Manuals	6
COPE Family of Products General Information Manual	6
COPE for IMS/DC Installation Guide.....	6
COPE for IMS/DC Administration Guide.....	6
COPE for IMS/DC Distribution Tape Information	7
COPE for IMS/DC Programmer's Guide	7
COPE for IMS/DC Messages Manual	7
Distribution of Manuals.....	7
New Features	8
MQM Support.....	8
FASTGEN Feature.....	8
Support Virtual Date for Different Logical Systems	8
Synchronized MFS MSC support.....	8
Support Version 5 of IMS	9
Support of TRIMAR Secondary Index Utility	9
Support Packages for DB2.....	9
ACB Generation Changed for STUBX Copy	9
COPE Modules Moved Above the Line	9
Scrolling COPE Menu.....	10
Support for the ViaSoft Development Tool.....	10
Concatenated DDnames and ESTAEs Displayed at Logon	10
Additional transactions added for PSB overflow.	11
Unconverted JCL Support of HSSR,IC Plus and Pointer Checker.....	11
Unconverted JCL Option.....	11

Load Module Date + AMBLIST Generation.....	12
Member Name Selection Mask Capability.....	12
Separate Message Region Support.....	12
Enhanced DB2 DBRM Support.....	12
DBSTOP and REFRESH Operations Added to Option 4.2.....	13
DBRC Support (Option 5.10.14).....	13
"T" Edit Command (Translates to C-numbers).....	13
Accounting Information.....	14
Note on MAC Type.....	14
Logging Users on En-mass.....	14
Backup/Restore.....	15
Custom IMS Commands.....	15
SUB Command under IMS.....	15
BMP IMS Commands.....	16
DBD Rename Information (Option 6.7).....	16
JOB Command for Checking and Resubmitting Jobs.....	16
Note on Databases Not Available (NA status).....	16
Selecting COPE Sub-options from the ISPF Primary Menu.....	17
Allocating COPE Libset Libraries.....	17
Specifying Unused Logical Systems.....	18
Call Trace Browse Filters (Option 2.7 or 7.2).....	18
Ready Trace to IMS Log.....	18
Compare and Import Option.....	19
Fast Path Support.....	19
External ACBGENs.....	19
Ctblks and Nucleus IMSGENs.....	19
CTL, DLS and DBRC Region JCL Notes.....	21
Message Region JCL Notes.....	22
Message Region Preload.....	24
Message Region Dataset Support (Open Intercept).....	26
BMP, DL/I and DBB Regions and COPERCO0.....	27
With COPERCO0.....	28

Without COPERCO0	28
Without COPERCO0 and with External PSBs	29
Without COPERCO0 and with COPE PSBs	29
Tuning IMS Systems.....	30
/FOR from Cleared Screen.....	30
Abends under IMS.....	31
4700 Tracing Support	32
Sort In-storage Option (COPE SORT).....	32
SYNC Command Improvements.....	33
Enhanced DYNALLOC and Stage 1 Processing (Common Specs.)	35
Adding Stage1 Statements - @COPY.....	35
Xpediter - COPE	36
Log Translate for MICS/IMF - COPEARCM.....	36
Advantages.....	36
Disadvantages.....	36
Zdefault Parameter Edit - EVARS.....	37
Support of BMC DELTA/IMS	38
Support of RECON extract for DYNALLOC.....	38
Support of DEDB Area Datasets.....	38
Support of RACF for ISPF.....	38
Compress/Expand PDS Utility (Option 3.8).....	38
Product/Feature Support.....	40
Table 1: Special Product Feature Support	40
Table 2: Products not needing special COPE support	41
Index.....	42

Preface

About This Manual

COPE for IMS/DC New Features Bulletin

Explains new features of the latest release of COPE, which may not yet be included in the appropriate manuals.

This manual is for individuals who require details about new features of the latest release of the COPE for IMS/DC system.

Related Manuals

COPE Family of Products General Information Manual

Gives an overview of COPE products (including COPE for IMS/DC) and explains the benefits to be gained by their use.

This manual is for individuals who wish to evaluate the use of COPE products.

COPE for IMS/DC Installation Guide

Explains how to install or upgrade a COPE for IMS/DC system. Details are given on unloading the distribution tape, setting up ISPF logons, and specifying system-related parameters. This manual should be read in conjunction with the *COPE for IMS/DC Distribution Tape Information*, which accompanies each distribution tape.

This manual is for individuals who are responsible for the installation or upgrade of a COPE for IMS/DC system.

COPE for IMS/DC Administration Guide

Explains how to create, administer, and run a COPE for IMS/DC system.

This manual is for a Database or System Administrator who is setting up a COPE for IMS/DC system. The reader should be familiar with IMS concepts, such as Program Specification Block (PSB) and Database Descriptor (DBD).

New Features

New features are listed below, with the newest added features at the top and the oldest added features listed last.

MQM Support

The COPEXUX exit has been extended to allow MQM calls to be intercepted and the queue names changed to a value that is determined by the Logical System the transaction is executing in. This allows different versions of application programs to execute simultaneously in the same environment and use different MQM queues without changing source code.

FASTGEN Feature

Installations which have a large number of statements in their STAGE 1 definitions, or which require a large number of Logical Systems will benefit from purchasing this additional feature. It radically reduces the time taken to perform PSB generations, Stage 1 Generations and REFRESH operations.

To effectively utilize the feature, additions to STAGE 1 definitions should be made via the 4.2.E;I option. This allows individual DATABASE, APPLCTN and TRANSACT statements to be added to individual, or groups of, Logical Systems. The additions are maintained in such a way as to eliminate duplicate processing during the option 4.2;GS STAGE 1 generation process.

If Stage 1 statements are to be deleted, maintenance of the STAGE 1 source should be performed via option 4.2;E;S or via the BMC DELTA support feature. Using option 4.2.E.S will increase the Stage 1 generation time, however it will be less than if the feature were not installed.

Support Virtual Date for Different Logical Systems

Every Logical System may use a different date from the System Date in the CPU. This is a separately priced feature of COPE and works with the ISOGON Tictoc product which must be separately installed.

Synchronized MFS MSC support

A separate option (4.12) has been added to synchronize the MFS MID/MOD names between separate physical COPE systems that are connected by MSC links. This ensures correct operation when a screen definition is generated in one physical system and displayed on a terminal connected to another physical system.

Support Version 5 of IMS

The DFSAOE00 Type 2 exit is used to allow command to be entered from any terminal and have the names translated to the database and PSB names that relate to a Logical System. The format of the command is **:/LOG /DIS DB <DBNAME> LSYS <LSYS-NAME>**. The /LOG is required to allow the exit to communicate with a long running task which issues the commands after the names have been translated.

Support of TRIMAR Secondary Index Utility

This utility requires a modified ACB generation procedure which is now supported and activated by the specification of a global variable.

Support Packages for DB2

COPE now can convert existing plans that consist of only DBRM's, or DBRM's and Packages, or Packages only, to COPE plans that use **only DBRM's** or plans that use **only packages**. The conversion process is controlled by a ZDEFAULT variable names XCOPEPK.

In addition, importing a single DBRM for rebinding a plan results in accessing the DB2 catalog for inclusion of all other DBRM's in the plan. This results in increased usability of the external interface.

ACB Generation Changed for STUBX Copy

Previous to release 4.0.4, an ACB generation copied STUBX modules to the active PGMLIB before a /MOD PREPARE ACBLIB had been issued. This produced an exposure of a mismatch between the active ACB and the description in the STUBX. The current release does not copy a STUBX unless a mismatch is detected in the online system. The copy is performed in the message region and is dynamic.

COPE Modules Moved Above the Line

COPE system modules now have been moved above the line for maximum use of space below the line for application modules and ISPF workspace. An alternate method of running Xpediter from a CLIST has been introduced to allow maximum space to be made available.

Scrolling COPE Menu

If the result of IMS commands exceeds a screen, e.g. the result produced with a /DIS DB STATUS IMS command, the output may be displayed by scrolling with PF7 and PF8. A 'Find' command is also implemented similar to that used within ISPF Edit.

Support for the ViaSoft Development Tool

An additional feature has been made available for support of the ViaSoft program development tool. The support includes an additional Stage 1 generation and instructions that enable ViaSoft and COPE to be used together.

Concatenated DDnames and ESTAEs Displayed at Logon

Release 3.9.9 changes the panels displayed when a user logs onto an Lsys, to display the DDnames used for program loading, and the ESTAEs currently active. Normally you will see these new lines:

The DDname search order is: C000000n COPESTEP STEPLIB
Estaes curr active: COPEXP7 DFSPCC20

This is **not** a change to the treatment of either program loading or ESTAEs, it is merely extra information intended to clarify the environment for users.

The implications of these messages are:

- In COPE message regions, the three DDnames C000000n, COPESTEP and STEPLIB are "concatenated", so that it is as if all the libs on the DDs were on a single STEPLIB. The order is C000000n first. This means:
 1. In the "DDname search order" display, C000000n is the Lsys-specific DDname used for loading programs in this msg region for this Lsys. If you change Lsys's, you will see a different DDname here (different Cnnnnnnn). The libraries come from ISPF 4.11 definition, via COPEXRF4. The Lsys-specific DDname does not apply to DFSMPLxx-preloaded modules (which are loaded from COPESTEP).
 2. COPESTEP is logically concatenated after C000000n. If your program loads a module that is not in the Lsys-specific concatenation, the next DD searched by MVS is the COPESTEP. Preloaded modules are loaded from COPESTEP (followed by STEPLIB, then LNKLSTxx).
 3. STEPLIB is logically concatenated after COPESTEP. If your program loads a module that is not in the Lsys-specific concatenation, nor in the COPESTEP concatenation, the next DD searched by MVS is the STEPLIB. Since in a COPE message region you should only have ONE library on STEPLIB -- the authorized library containing COPERC00 -- none of your loads

should be resolved from STEPLIB.

4. LINKLSTxx libraries are logically concatenated after STEPLIB, as in all MVS regions. If your program loads a module that is not in the Lsys-specific concatenation, nor in the COPESTEP concatenation, nor in the STEPLIB concatenation, the next DD searched by MVS is the STEPLIB.

The term *Tasklib Switching* as used in the COPE manuals refers to the way COPE dynamically changes the Lsys-specific DDname (the top C000000n one) to the appropriate Lsys, for each transaction that runs in a COPE message region. This is the essence of shared (by Lsys's) message region support in COPE.

ESTAEs are displayed to assist with problems where your application program either leaves ESTAEs active in the message region, or cancels other module's ESTAEs. To diagnose such a problem, you would issue the LIB command (which displays the same screen as when you logon to an Lsys). The screen will show you the current ESTAEs in this message region. If all is well, you should see COPESP7 and DFSPCC20 (only) as above. We are not implying here that ESTAEs are related to program library concatenations -- these are two separate unrelated issues.

Additional transactions added for PSB overflow.

If a COPE PSB exceeds 255 PCB's or 1000 SENSEG statements, an overflow PSB is generated. Before Release 3.9.3, a Stage 1 generation was required to allow the generated overflow transaction and PSB to be defined to the COPE physical system. COPE now automatically adds a set of transactions that are reserved for overflow usage, and will assign them as necessary when overflow transactions are required. This eliminates the requirement to do a Stage 1 modblks generation, and the availability of the COPE system is increased.

Unconverted JCL Support of HSSR,IC Plus and Pointer Checker

The unconverted JCL option now translates the control cards for the above products.

Unconverted JCL Option

Option 4.9 now supports the Unconverted JCL Option. This option eliminates the requirement that batch and BMP JCL be converted to use the COPE generated DBD's/PSB's and Dynalloc information. The user has to identify the Lsys in the JOB card or EXEC card and all name changing is performed dynamically when the job is executed.

Load Module Date + AMBLIST Generation

Entering **LOAD** on the primary COPE menu invokes an application that allows the dates of load modules to be extracted and displayed. Members may be selected and AMBLIST JCL is generated and submitted to background.

Member Name Selection Mask Capability

Anywhere that a member name may be specified, a partial name with special characters in it can be entered to cause the resulting display to only contain members that match the mask. e.g. Entering a name of *a+ results in a list of members that have the last but one character of 'A'.

In addition, the **S** capability on all displays has been enhanced to allow multiple masks to be specified e.g.

```
s a* *b a*C*
```

This selection statement selects all members that begin with an A or B and which do not have the character C anywhere in the name. Entering a **S** without a mask results in a prompt display that describes the masking characters.

Separate Message Region Support

Option 4.2.SEPREGN allows definition of Psb's that use different message regions per Lsys to process transactions in. This feature is intended to allow extensive preloaded modules with different versions per Lsys to be tested. The existing COPE preload facilities are not adequate in certain installations due to limitations on the amount of message region memory.

Enhanced DB2 DBRM Support

Option 5.7 allows definition of a COPE DBRMLIB and a dataset external to COPE that contains DBRM modules. Any number of DBRM modules may be selected and assigned to a Plan, or modules may be selected and all plans that reference them are rebound. The enhancement is intended to allow JCL procedures currently in use to automatically import DBRM members and have COPE assign the correct internal name and perform a bind.

DBSTOP and REFRESH Operations Added to Option 4.2

Option 4.2 allows access to the DBSTOP application with which databases may be defined as having an initial stopped status.

The REFRESH application allows regeneration of all x-reference members and also the data that reloads the COPE control database. It automatically submits the jobs the refresh all the data.

DBRC Support (Option 5.10.14)

Option 5.10.14 uses the information extracted from the DBDs and derived groups of databases, as well as dataset names extracted from the DYNALLOC source, and generates DBRC INIT.DB, INIT.DBDB and INIT.DBGROUP statements for database definition to DBRC.

The generated statements should be modified in EDIT, to reflect the installation's DBRC parameters.

"T" Edit Command (Translates to C-numbers)

The "T" edit command has been added to the JCL generation process. This edit command is intended to be used when utility control cards are required to be changed to reflect the COPE C-number names for DBDs, DDnames, PSBs, and Programs. The control cards should be imported using the option 4.9 application and the members accessed with the "ES" command. A description of the "T" command can be obtained by pressing <PF1> (HELP) and reviewing the tutorial.

Accounting Information

You can obtain accounting information by logical system, using new features of the archive exit, COPEARCX. Please see the COPE for IMS/DC *Administration Guide* for a full description of the Archive Exit.

Note on MAC Type

A MAC type with a PDS indicator of "N" must be defined via Option 4.1.4. The COPE for IMS/DC *Administration Guide* should be altered to reflect the addition.

Logging Users on En-mass

Users can be logged-on en-mass from a batch job by using the following JCL:

```
//LOGONEXEC PGM=DFSRRC00,REGION=2048K,  
//PARM='BMP,COPEIMS3,COPEIMS1,,,,,,,,,IMD1'  
//STEPLIB DD DSN=TFTPROD.COPE.LOAD,DISP=SHR  
//DD DSN=IMSVS.TEST2.RESLIB,DISP=SHR  
//SYSPRINT DD SYSOUT=*  
COPE MACPACD1 FROM #TS1002  
COPE MACPACD7 FROM #TS1003  
COPE MACPACD1 FROM #TS1004  
... ETC ..
```

You can also use a new LOGOFF FROM ALL command from a BMP or a terminal to logoff all users.

Note: You must use LOGOFF FROM ALL after you add or delete logical systems, or activate or inactivate them. If you don't, those that are logged on will get library allocation failure messages.

You can also xxxxx FROM ALL to log all uses onto a particular system xxxxx (but you should LOGOFF FROM ALL them first). The "FROM ALL" commands are slow, because they scan USTDLMGR.

The above JCL can be used to insert any COPE or IMS commands from a BMP step. You use this in BACKUP/RESTORE jobs to issue COPE MSG commands back to the user, to tell them when the job has started and ended, and whether the condition codes are good or not. See JCL(RJOB1) for a detailed example of a RESTORE job.

Backup/Restore

If you have users who need to be able to backup and restore databases easily from their IMS terminals, you can set up BACKUP and RESTORE commands. This feature enables you to:

- Give some users (e.g. Demonstration System users) IMS-only access
- Save users the trouble of logging on to ISPF to backup/restore
- Tailor the backup/restore jobs, for dataset names, volsers, etc
- Give users feedback messages about backup/restore jobs

The delivered BACKUP/RESTORE commands will not work unless suitable members are created in the JCL library pointed to by the message region COPEJCL DD. You may also want to vary the members to be submitted, by modifying the COPEO member of the MENUS library.

Please see the *COPE for IMS/DC Administration Guide* for a full description of Backup/Restore.

Custom IMS Commands

You can set up your own command to be a sequence of existing COPE or IMS commands, by editing the COPEO member of MENUS, and adding a lines as in these examples:

```
DISQ, '/DIS Q TRAN;/DIS A'  
DISNODE, '/DIS NODE DFTI2L*;/DIS NODE DFTI4L*;/DIS MASTER'  
STOPALL, '/STO DB TESTSYS;/STO DB DEMOSYS;/STO DB PRODSYS'  
BOUNCE, '/STO REG &TPREG;/STA REG &TPJOB'
```

New (in release 3.7.9) &-variables &TPREG and &TPJOB are provided for the BOUNCE command, so that you can stop and start the message regions without looking up the ids and job names using /DIS A.

SUB Command under IMS

The SUB command (used in BACKUP and RESTORE) submits jobs, substituting variables with support for ISPF-style)SEL,)ENDSEL,)SET, and)CM statements. See member JCL(RJOB1) for a detailed example of a RESTORE job.

Please see the *COPE for IMS/DC Administration Guide* for a full description of the SUB command.

BMP IMS Commands

COPE or IMS commands can be executed from a batch (BMP) job by using the COPEIMS3 program (which uses COPEIMS1 PSB), for example:

```
//COMMAND1 EXEC PGM=DFSRRRC00,  
//PARM='BMP,COPEIMS3,COPEIMS1,,,,,,,,,&XIMSID'  
//STEPLIB DD DSN=TFTPROD.COPE.LOAD,DISP=SHR  
//DD DSN=IMSVS.TEST2.RESLIB,DISP=SHR  
//SYSPRINT DD SYSOUT=*  
COPE TESTSYS FROM DFTI9L02  
COPE MSG &ZUSER SEND MYSELF A MESSAGE  
COPE STO DB DEMOSYS  
/*
```

DBD Rename Information (Option 6.7)

Option 6.7 displays one or all of the logical system's DBD name conversion table. The information is an extract from the DBD table with included DBDs added and modified with excluded DBDs.

JOB Command for Checking and Resubmitting Jobs

The new JOB command is a powerful feature. You can access any generated job JCL and either modify it with Edit, or resubmit it for repeat processing. If a job is very large and fails because of CPU time or DASD space, you can split it into several jobs using Edit, and run each job separately. It is usually obvious how to subdivide a generated job, since it usually consists of JCL and recognizable member names.

Note on Databases Not Available (NA status)

The way COPE acts if a database is not available is controlled by the HANDLENA specification. The default is to send the user a message telling them that the database is unavailable, and then not to run the program (HANDLENA=N). You can change this so that the program runs (HANDLENA=Y) even when there are databases unavailable in this logical system. The attribute tells Cope whether the program "handles NA", or not -- most programs aren't coded to test for NA.

The HANDLENA specification is entered via the 2.5 option (Specification of Program I/O data) and is documented in the *COPE for IMS/DC Programmer's Guide*. To specify the attribute for a program, go into option 2.5 for the PSB source that corresponds to the program. Alternatively, in the case where you import the program into Cope, you can specify the HANDLENA option in 2.5 for the program source

or load. For a new HANDLENA value to take effect, a PSB and ACB generation must be performed. This is because the specified value is copied into the STUBX module generated when a PSBGEN in 4.7 is performed. The STUBX module is copied to the online system when you ACBGEN from 4.8.

Selecting COPE Sub-options from the ISPF Primary Menu

If you desire to select COPE sub-options from your primary panel by entering X.Y.Z - you should add the following statements to your primary option menu:

```
&A = TRANS ( TRUNC(&ZCMD,') *,*)
&NEXTOPU=.TRAIL
&ZSEL = TRANS (TRUNC (&ZCMD,')
0,'PANEL(ISPOPTA)'
1,'PGM(ISRBO)'
*
*
      COPE,'PGM(COPEINIT) PARM(ZDEFAULT(member-name)+
DZENV(F) PROCLIB(proclib-name)+
NEXTOPU(&NEXTOPU)) NEWPOOL NEWAPPL(COPE) NOCHECK'
**
```

These statements pass trailing characters to COPE while creating new shared and new profile pools. To exit COPE, use "=X".

Allocating COPE Libset Libraries

A new method has been implemented to allocate new COPE Libset libraries. Option 3.2.DA now displays two rowtypes when it is selected. The first rowtype contains the datasets and their LRECL, BLKSIZE, VOLUME, RECFM and UNITS attributes. The second rowtype allows specification of the primary and secondary and directory block attributes for every defined library. If a dataset is shared between COPE libraries, the space and directory definitions are accumulated before the library is allocated on DASD. The *COPE for IMS/DC Administration Guide* should be marked to indicate that the screens look different (the data is split across two screens). The VOL, MULTIPLY, PRI and SEC commands still operate as described in the guide.

Specifying Unused Logical Systems

You can specify logical systems to be used at a later date (unused currently). This will improve performance of dynamic allocation in large systems. This means you can afford to pre-specify many "spare" logical systems that may be populated at a later date.

You set the "Active Flag" Y or N indicator when you do a COPE definition in option 4.2.E.S.C. For IMS release 1.3, you should specify all systems as "Active" Y (this facility works only for IMS 2.0 and higher).

Call Trace Browse Filters (Option 2.7 or 7.2)

Call Trace Browse now selects transactions or terminals using * "wild" characters, which can be anywhere in the mask (like ISPF member selection masks). Please see the *COPE for IMS/DC Programmers Guide*, TRACE section, for a description of the display, or hit <PF1> while in option 2.7 or 7.2.

Ready Trace to IMS Log

Program traces, such as Cobol Ready Trace now go to the IMS log, instead of to the JES2 queue, and are browsed in ISPF COPE option 7.2, in the same way as DL/I and SQL call trace. Due to buffering, individual lines may be displaced slightly from their correct positions. From a single module, the lines come out in the correct order.

All output from an MPP to a DD in the message region assigned to SYSOUT is intercepted and sent to the IMS log. The output of programs cannot be browsed via IOF or SDSF, you must use COPE option 7.2 to view this output on the IMS log.

This feature is active whether or not you have TRACE ON or OFF. To turn off this feature (in an emergency) you must turn it off for all logical systems, using DEBUG MODE 5.

Please see the *COPE for IMS/DC Programmers Guide*, TRACE section, for a description of the display, or hit <PF1> while in option 2.7 or 7.2.

Compare and Import Option

A new compare and import option has been added to COPE option 3.7.

You can now compare two datasets and submit a generated job that will import members that are different, and/or members that exist in one dataset but which are missing from the other dataset. The compare reads the complete member text for identically named members and produces a hash number. If the hash values differ, the members are different.

COPE Option 3.7 also allows members in a COPE library to be compared with members in an external dataset. A job is generated that will synchronize the information in the COPE and external datasets by importing information from the external dataset into COPE for any members which are missing or which have hash numbers that are not identical. Members that are in COPE but which are not in the external dataset can be optionally deleted.

Fast Path Support

Support of Fast Path IFP regions has been added.

You can now convert IFP-type BMP JCL in option 4.9 for Fast Path message regions. These are wait-for-input message programs. Import existing Fast Path message region JCL using option 4.9 sub-option "I", parse it using sub-option "P", and then regenerate it using sub-option "G". The resulting converted JCL should be executed in place of the original JCL.

External ACBGENs

An interface to support external ACBGEN's has been added.

If an ACBGEN is required to be invoked from outside COPE, the correct translation of PSB and DBD names must be performed. An interface has been added to 5.7, where you specify PSB or DBD names, and generate a job that will generate the correct control cards for the ACB processor. An explanation of the interface is in the COPEJCL sample JCL that is generated with the GP option of 5.7.

Ctlblks and Nucleus IMSGENs

Option 4.2.GS presents you with a screen where you can select the IMSGEN type, Modblks, Ctlblks or Nucleus. Prior to release 3.7 of COPE, only Modblks was supported. Now Ctlblks and Nucleus gens are supported. The JCL that is generated, executes a proc, which first checks the Stage 1, then does the Stage 1 assembly. The output of the Stage 1 assembly runs into a program that splits out the assembler and linkage editor cards, and puts them in datasets suitable for running into a single large Stage 2 assembly in the same job.

The Stage 2 assembly is able to do multiple assemblies in one step, because it uses the BATCH assembler option, on the parm. The links, security gen. and JCLIN are likewise all done in the same job.

The advantage of this method for doing MSGENs is that the entire gen. runs in one job, with good checking, leading to fewer possibilities for error.

To use this MSGEN method successfully, you must set the ZDEFAULT dataset name prefixes correctly (for example, XCOPESMP is the prefix of your GENLIBs, IMS source libs, etc, XCOPEOPT is the prefix of your OPTIONS dataset).

In some cases, you may find it necessary to do minor JCL tweaking to the JCL procs supplied. The procs are: COPEMGEN (Modblks gen.), COPEGEN (Ctlblks gen.), COPENGEN (Nucleus gen.) and COPENGE2 (Nucleus gen. for IMS release 2 and below). An example of the tweaking that may be required is if you have your SMP distribution library prefixes different from you target library prefixes. In this case, you could let the COPESMP proc parameter be your target libraries (MACLIB, etc), and hardcode the DSNs for your distribution libraries in the procs. Note that the COPESMP proc parameter corresponds to the XCOPESMP ZDEFAULT parameter (with an X on the front).

For documentation on the mechanisms used in the three programs that are used in this method -- COPEGEN1 (Stage 1 checker) COPEGEN2 (Stage 2 splitter) and COPEGEN3 (MODSTAT flipper/BMP caller) -- see the prolog in the ASM source for these three programs.

For Ctlblks and Nucleus gens (and not for Modblks gens), a new COPEXOVR step is generated on the end of the MSGJCL. This proc executes some "Cross-over" steps, and the SMP JCLIN. The crossover puts the result of the gen. into your run RESLIB, and SMP RESLIB. The previous steps have only updated a COPE RESLIB called RESLIBN (for RESLIB New).

The cross-over won't run if IMS is up. It first tries to flip MODSTAT DISP=OLD. If it can't get DISP=OLD, it assumes IMS is up, gives a cond code 4, and the subsequent steps won't run.

You

should be aware that you must not cross-over a Ctlblks or Nucleus gen. if IMS is up (because IMS may need to emergency restart). The mechanism stops this from happening. You also must be prepared to always cold start the next time you bring IMS up after a cross-over of a new gen.

The cross-over backs up your current run RESLIB to another COPE RESLIB called RESLIBO (for RESLIB Old). It then copies RESLIBN (New) into XCOPEIRS (your run RESLIB) and XCOPEIMP.RESLIB (your SMP RESLIB). The two COPE RESLIBs, Old and New, are only there for the purposes of backup and staging, and should be ignored (never used) outside the MSGEN environment. Allocation for these RESLIBs is provided in the ALLOCIMS member of JCL. Old is full-size, because it is a full backup, whereas New is small, because it only contains the output of the gen.

These gen. processes assume that you allocate the datasets as in ALLOCIMS. In particular, everything (except RESLIB) in the online run-environment has the same prefix that you specify in XCOPEONL. This is usually the most desirable way to run -- but you may need to tailor this for your own special dataset-naming considerations. The gen. process as delivered is reliable and comprehensive (for example, space failures are avoided by compressing at appropriate points).

CTL, DLS and DBRC Region JCL Notes

CTL Region JCL needs an extra COPEPGM DD pointing to the XCOPEPGM library (STUBX and XRF module library). The COPE AOI exit, DFSAOUE0, reads (does not load) the translation modules COPEXRF1 and COPEMFSX from this COPEPGM DD. The fact that it reads rather than loading means that the COPEPGM DD library does not need to be (and should not be) authorized.

The DB2 SSM parameter can be specified in the CTL region parm (49th parm, 48 commas in), or in the Message or BMP regions (14th parm, 13 commas in). In releases of COPE prior to 3.7.9, there was a requirement to only specify this in the Control Region. This is no longer necessary.

The DB2 SSM member should be coded as in this example:

```
DSNF,SYS1,DSNMIN10,,,-
```

In other words, there is no special COPE requirement here, except that the second parameter, the SQL Id must match the COPE zdefault variable XCOPESQL. If it doesn't match, you will get a U3042 abend.

To fix a SQL Id mismatch, fix the XCOPESQL variable (for example, use the new hidden option EVARS from the primary COPE menu), then run 4.2.MDS, then stop and start your message regions (the new BOUNCE command will do this).

Prior to COPE release 3.7.9, there was a requirement to put in the SSM an RTT table called COPERTT. This is no longer needed, and will not be used if coded, so remove it if you have it specified.

The LSO=S option must be specified in the control region parm (45th parameter, 44 commas in). This causes a DLS ("DLI Separate Address Space", or "DLISAS") region to be used, and the DLS region JCL must be set up, with ACBLIBs matching the Control Region exactly. If you do not specify LSO=S, COPE will consume large amounts of CSA (many times larger than in a non-COPE system), and this will mean that you can only run as many logical systems as you would be able to run non-COPE systems, on one machine. This defeats the main purpose of installing COPE. However, if you code LSO=S, CSA per logical system is only about 100K, after an initial amount of about 1M.

Other than these requirements, CTL, DLS and DBRC region JCL is the same as in a non-COPE system.

Examples are provided in the JCL library, members:

- COPECTL
- COPEDLS
- COPEDBRC

Message Region JCL Notes

You must set up message region JCL as follows:

- EXEC PGM=COPERC00
- The PARM= 4th parameter (3 commas in) must be the xx that you assigned to your DFSMPLxx (preload) member.
- Specify 99 as the 7th parameter (6 commas in). This is the IMS BLDL option. Application programs are not loaded using this BLDL so 99 helps performance (of STUBX loading) while still letting you pick up new versions of application programs on the fly.
- You can specify an SSM parameter if you want to in the message region (14th parameter, 13 commas in). The SSM member must have a line with the second parameter (the SQL Id) matching the new zdefault XCOPESQL variable.
- //STEPLIB DD to the authorized library containing COPERC00 (the one named in the XCOPEAUT ZDEFAULT variable). Do **not** code any other library (even RESLIB) on the STEPLIB.

The only exception to this is a product called Table Base, which due to its non-standard alteration to the MVS task structure, must be authorized, and on the STEPLIB.

- //COPESTEP DD to:
 1. XCOPEPGM (this is the STUBX library)
 2. XCOPE1.LOAD (the COPE software load library)
 3. Any library that you would normally code on the STEPLIB of a message region, that you want to appear in **every** logical system's concatenation. For example, the Cobol run-time library. Do not code libraries that you want to appear in one logical system, but not in others.
 4. XCOPEIRS (your run RESLIB)

Message Region Preload

To vary preloaded programs across logical systems, you import them into COPE, even if Tasklib Switching is being used. When the program is imported into COPE it is renamed to a C-number name. The C-number name needs to go in the preload list (DFSMPLEX member).

Option 5.10.15 has been added. In option 5.10.15, the COPE Administrator lists the load modules for the DFSMPLEX preload list in the message region. The utility is a convenience that removes the necessity of discovering the C-number names by accessing option 6 (the COPE Dictionary).

In release 3.6 and before of COPE, if you specified Tasklib Switching, there was extra overhead in invoking preloaded MPPs because of the "Dynamic STUBX" module, or "Dynx". Also, the program was deleted from the message region on a logical system change, by COPE cleanup. The COPE processor in the message region has been changed to access preloaded MPPs directly, bypassing the Dynx module, to improve performance. COPE cleanup won't delete the C-number programs from the message region, because they are now directly preloaded.

Preloading is often necessary for modules that open datasets and/or save addresses of data in storage.

To display the preload list online, use VERSION LLE. This displays the MVS Load List Elements, and will show modules that are in storage, either preloaded, or loaded and not deleted by application programs. If the display of VERSION LLE exceeds one screen, re-issue specifying VERSION LLE <start>, where <start> is the first module name that you want the sorted display to start from (can be a partial start name).

An example of the output from VERSION LLE is:

```
COPE TFT4DFTW1S0121:57 MSGCOPEA J4137
```

```
====>
```

```
CACEMCKI 1 1 CBLTDLI 1 1 COPEXP1 1 1 COPEUTP1 1 1  
DMLOGIC 1E 1 DMTIMEOU 0E 0 FMTALIAS 0E 0 ILBOACS0 1 1  
IMS0260* 5
```

Nameij

i is number loaded at region startup

E indicates Exempt from cleanup

* indicates module would be deleted by Cope cleanup on lsys change

j is current number loaded

You may need to take special action for modules that show up as **0*** (IMS026 in the example above). The **0** means that these modules were not preloaded. The ***** means that they would be deleted on a logical system change by COPE in its cleanup. Normally, this cleanup action is OK.

However, sometimes, the module is designed to remain in the region, with pointers to it in preloaded modules or getmained storage. This is incompatible with the cleanup. Programs will S0C4, when they follow the address pointer and find that COPE has deleted the module. If you have such modules, they must be preloaded, to avoid these S0C4's. You can either preload the module, or import it and preload it. If you preload without importing, the same version of the module will be used

across logical systems. If you import and preload, it will be a Cnnnnnnn module, and different versions will be supported across different logical systems.

In some cases, particularly where assembler routines call Cobol subroutines which have incompatible ENDJOB parameters, preloading still gives SOC4's. In those cases, you must make the modules Exempt from cleanup. To do this, change the list of exempt modules in COPESXP2 at label EXEMP. This is a sorted list. Assemble and relink COPESXP2, using JCL member ASM as a guide. In the above example, DMTIMEOU and FMTALIAS are exempt from cleanup, because they contain Cobol subroutines that must remain in storage, but cannot tolerate preload. DMLOGIC is both preloaded and exempt, the exempt status in this case being redundant.

You can also check for cleanup problems, by looking on the message region job log. Every time a module is cleaned-up, there are messages such as:

```
COPESXP2 MODULE NOT ON SAVED LLE=IMS026  
COPESXP2 DELETE IMS026 5 TIMES. NOT PRELOADED
```

COPE cleanup only deletes modules, it does not freemain storage. Use DEBUG REGION to display the largest free storage in the region, to check that cleaned-up modules are not leaving getmained storage around.

Message Region Dataset Support (Open Intercept)

Some programs in message regions access non-IMS disk datasets. If the disk dataset needs to be different in different logical systems, you can use a new COPE "Open Intercept" mechanism to make them different.

The mechanism dynamically varies the DD name in the DCB or ACB at OPEN time. This will work for VSAM files, sequential files and PDSs, including loading modules from PDSs.

To vary the datasets used in message regions, you add additional rows in ISPF option 4.1.5 (formerly this option was 4.1.6). In the rows you give the DD and DSN for each logical system (can be a concatenation of DSNs). A DSN can have a member name in parentheses on the end (so the member is treated as a sequential dataset).

You optionally supply a "Modified DD", or "COPEDD", which is the DDname that COPE will change the "real" DD to, in the DCB or ACB. If you omit the COPEDD, COPE will assign a C-number name.

In option 4.1.5, the COMPRTN column is not used in this release of COPE (set it to blank) -- it will be used in a later release to implement an optional dataset compression feature.

After exiting option 4.1.5, you do an option 4.2.GS, and run the XREFJCL job generated by option 4.2.GS. This places the information in the COPEXRF4 module, loaded by the COPE processor in the message region. To force a refresh of the COPEXRF4 information, without having to bring message regions down and up, do a REFRESH command.

When your program in the message region opens a DCB or ACB that contains a DDname that you specified in option 4.1.5, COPE intercepts the open, and replaces the DDname in the DCB or ACB with the COPEDD. The COPEDD depends on which logical system is currently active in the message region. COPE also dynamically allocates DISP=SHR the dataset (or concatenation of datasets) whose DSNs you also specified in option 4.1.5. COPE then passes control to MVS open. When your program receives control back from open, it can then read, write, or load from the file. If the same program runs in a different logical system, it will get a different file.

Notice that you specify the DSNs for the message region datasets in option 4.1.5, and you do not code them in the message region JCL. COPE uses dynamic allocation for these datasets, so that there are no availability problems, due to message regions holding onto datasets. If you need to allocate DISP=OLD to a dataset that is used by the message region in a batch job, you make the message region free the dataset by stopping the logical system "on the TR side" (i.e. SS, then P, TR and logical system name). This will deallocate all datasets that have been dynamically allocated by that logical system. There may be a slight delay in this deallocation -- it actually occurs on the first transaction through each message region immediately following your issuing of the stop (P) command.

This use of dynamic allocation and deallocation avoids the availability problem that would occur (in other logical systems) if these datasets were hard coded in message region JCL. If there was no deallocation capability, you would have to bring down the message regions. It is for the same availability reason that you specify STEPLIB-type program libraries in option 4.1.5 (called TASKLIBs), and they are dynamically allocated, rather than hard-coding them in the message region JCL.

The following considerations apply to message region datasets:

- The program that opens DCBs or ACBs may or may not need to be preloaded. It needs to be preloaded if it leaves the DCB or ACB open across transactions. If it opens and closes in one transaction, it does not need to be preloaded.
- If the program obtains the DDname after open, it will find the COPEDD, and not the original DD that it specified in the DCB or ACB, before open. If the code is sensitive to the DDname after open, you may need to specify COPEDDs that satisfy this sensitivity, in option 4.1.5.

The option 4.1.5 table format (formerly option 4.1.6) has been changed. To specify the program libraries for each logical system, you now code rows with TASKLIB in the DD column. The TASKLIB is not a real DD, it is MVS terminology for a set of libraries equivalent to the STEPLIB (in fact the TASKLIB libraries are concatenated ahead of the STEPLIB). For datasets that you wish to vary across logical systems, you code rows with the DDname column value the same as the DDname that in a non-COPE system, you would code in the message region JCL. In the COPE system, you code them in 4.1.5 instead of in the message region JCL. Do not code them in both places, as this would defeat the purpose.

When you code your message region JCL, you must omit the program libraries and/or other datasets that you put in option 4.1.5. In the message region JCL, you put the COPERCO0 authorized library on the STEPLIB. Do not put any other library on the STEPLIB, for example, do not put RESLIB on the STEPLIB.

The only module that will ever be loaded from the STEPLIB is COPERCO0. You put COPE's XCOPEPGM library (i.e. STUBX library), COPE's LOAD library, your RESLIB, and your program libraries that are common to all logical systems on the COPESTEP. The COPESTEP is an "unauthorized STEPLIB", common to all logical systems. Typically, COPESTEP common libraries include such things as the Cobol run-time library. The libraries on the COPESTEP are concatenated after the TASKLIB libraries you specify in option 4.1.5, and together these will appear to all programs the same as a STEPLIB appears to programs in a non-COPE system.

BMP, DL/1 and DBB Regions and COPERCO0

You can run BMP regions in two ways, and DL/1 and DBB regions in three ways:

1. With COPERCO0 (the COPE region controller)
2. Without COPERCO0 and with External PSBs (does not apply to BMP)
3. Without COPERCO0 and using COPE PSBs

MSG and IFP regions must be run under COPERCO0. CTL, DLS and DBRC regions must run without COPERCO0.

The third option above is for compatibility only with previous COPE releases.

With COPERCO0

This form is required if any of the following apply:

- BMPs are sensitive to the database names in the PCBs
- You want COPE DL/1 and SQL call traces
- You need DBRC forward recovery

Without COPERCO0

If none of the above apply, you can run DL/1 and DBB regions without COPERCO0, and using External (non-COPE) PSBs and DBDs. This means that you do not have to use COPE option 4.9 to convert the JCL.

With COPERCO0

You must convert JCL using ISPF option 4.9. You should set the new ZDEFAULT system variable XCDLIBMP to YES, and answer YES to the question "Do you want the COPE Region Controller", on the 4.9 screen, otherwise you won't get COPERCO0. The conversion does the following:

- EXEC PGM=COPERCO0 instead of DFSRRC00. Note that COPERCO0 calls DFSRRC00 (by attaching it), and therefore the IMS system environment is unchanged. COPERCO0 merely renames databases in the PCBs passed to programs, and provides optional call traces.
- The PSB name in the 3rd parm position changes to a Cnnnnnnn name. The program name in the 2nd parm position does not change.
- The STEPLIB points only to the authorized library containing COPERCO0 (the library named in the XCOPEAUT system variable).
- What was on the STEPLIB, including RESLIB, goes on the COPESTEP DD, which appears to the program exactly like a STEPLIB.
- The XCOPESTG (Staging STUBX library) is inserted on the COPESTEP, as well as XCOPE1.LOAD (COPE software load library).
- For DL/1 and DBB, (but not for BMP), a DFSRESLB is checked for and inserted, pointing to XCOPEIRS RESLIB.
- For DL/1, the COPE PSBLIB and DBDLIB are put on the IMS DD.
- For DBB, the COPE ACBLIB is put on the IMSACB DD.
- If database DDs are present, their DD names (not their DSNs) are renamed to the Cnnnnnnn names.

To get a DL/I and SQL call trace, you add:

```
//COPETRAC          DD SYSOUT=*
```

To turn off the call trace, either remove the COPETRAC DD, or code it DD DUMMY. Call trace comes out on the COPETRAC DD and is exactly as described in 2.7 when you press <PF1>, or in online IMS <PF1> then **U;2;7**, or in the COPE for IMS/DC *Programmers Guide*. The trace is a LOGIC trace, so it includes ===== lines that show the PCB list at the beginning, and COPE rename actions as they occur. The I/O area trace is truncated at 132 bytes, and there is no NOTRUNC option.

Without COPERCO0 and with External PSBs

This only applies to DL/I and DBB batch regions (not to BMP). There is no 4.9 conversion of JCL. You run with exactly the same JCL as under a non-COPE system, using PSBLIB, DBDLIB and ACBLIB that you must maintain external to COPE and in synchronization with COPE. The keeping of such libraries consumes some disk space. Synchronization is normally not a problem, if you insert the COPE PSB/DBD/ACB import and gen. steps into your standard JCL generation procedures.

You cannot use DBRC forward recovery, because the online will need to use different RECONS from the batch, and this means that DBRC forward recoveries could not pick up the complete list of logs from both environments. In a test environment that never uses DBRC forward recovery, this is not a problem. If you need DBRC forward recovery, then DLI and DBB must use COPERCO0.

There is no COPE Call trace if you do not run under COPERCO0, because it is COPERCO0 that sets up the call intercept.

Without COPERCO0 and with COPE PSBs

This option is provided only for compatibility with previous COPE releases. To get this method of operation, code ZDEFAULT system variable XCBMPDLI = NO, or answer the 4.9 "COPE Region Controller" question as NO. JCL must be converted in option 4.9. The conversion is similar to that described above, except that COPERCO0 and COPESTEP are not inserted. This method will work if the BMP, DL/I or DBB program is not sensitive to the database names in the PCBs. DBRC forward recovery will also work correctly.

There is no COPE Call trace if you do not run under COPERCO0, because it is COPERCO0 that sets up the call intercept.

Tuning IMS Systems

Tuning information is provided in the COPE for IMS/DC *Administration Guide*. The major points to be aware of are:

- Reduce IMS checkpoints
- Place datasets so that I/O is spread
- Use D/C monitor
- Tune pool sizes

/FOR from Cleared Screen

COPE can translate /FOR commands from a cleared screen, as well as /FOR commands from the COPE screen. If you need this feature, you should add the following command to your COLDS, WARMS and EMERS definitions in COPEO in MENU:

```
LOGON ECSA
```

This causes COPE to do extra processing at IMS startup, that scans USTDLMGR, and places Lterm/Sign-on to Lsys logon information in storage in Extended CSA, which is then used by the COPE /FOR intercept in the Control Region to do the translation of the modname.

The LOGON ECSA is necessary after an MVS IPL, because COPE's table in ECSA is not retained across MVS IPLs. It is therefore appropriate to have LOGON ECSA issued after every IMS startup automatically, in case there was an IPL. LOGON ECSA can take a minute or two to process, because it scans the entire USTDLMGR database.

If you do not require this /FOR translation, you do not need to do the LOGON ECSA. This is the reason that the COPEO member is delivered without LOGON ECSA -- it avoids the overhead at IMS startup for installations that do not require /FOR translation from a cleared screen. However, this may confuse users of the /FOR command, because they will get translation if they happened to logon to a logical system since the last IPL, but they won't get translation if they did not. The way around this is to either tell users always to use COPE /FOR (which always translates), or to tell users to always logon to a logical system (at least once per session), or to put the LOGON ECSA in CO- PEO.

Please don't confuse logging on to a logical system with LOGON ECSA. When a user logs on to a logical system, their single entry in the table is always created or updated. When a special LOGON ECSA command is issued, the entire table is recreated from all the logon records in USTDLMGR. The former is quick, and used by all. The latter is slow, and should only be used at startup of IMS, via the COPEO member.

If you are planning to use 4700 support, you will need the LOGON ECSA at IMS startup, because the ECSA table is used for the 4700 translation.

/FOR translation uses the COPEMFSX module in the Control Region. When you compile a completely new MFS, the MFSGEN job issues a COPE REFRESH command, which refreshes the copy of the MFSX module in both the Control Region and the Message regions. You should ensure that your MFSGEN refresh step (it is a BMP COPEIMS3 step) runs on the same CPU as your IMS system, or this won't work.

Abends under IMS

If an abend occurs on a particular transaction in the message region, hit PA1 to get the Abend Summary Screen. If this does not immediately show you the cause of the problem, issue a TRACE ON LOGIC command, to turn on the Application-level trace to the IMS log. Start the Tran (STA tran, or use SS screens), then recreate the problem. View this trace with ISPF option 2.7. This will quickly show such conditions as bad IMS status codes, with explanatory messages.

If COPE under IMS gets an abend, such as an S0C4 in a COPESXPn module in the message region, or an abend in DFSAOUE0 in the control region, you should issue the COPE DEBUG ON command, and recreate the problem. This produces a detailed trace on the COPERRMS DD in the message region, and on the Joblog in the control region. Recreate the problem with this trace on, then send the last part of the trace to Compuware. Also check the message region joblog for messages from COPE proceeding the abend. This greatly assists with solving the problem.

Don't use DEBUG ON for problems that are clearly in the Application program, and not in COPE, as the overhead is high. For problems that are in COPE (i.e. in a COPE module) you can use both TRACE and DEBUG (they can be both turned on at once).

For detailed information about these facilities, see the ISPF or IMS tutorial, which are identical, except for the initial entry point into the tutorial. Hit PF1 in IMS or ISPF, and follow the Help panel hierarchy. TRACE is described in **O** (Online), **9** (Trace), from ISPF, or PF1 then **9** from IMS. Trace output is described in detail in the Help in ISPF **2;7**, or in IMS Help **U;2;7**. DEBUG is described in **O;12;2** from ISPF Help. In IMS, you hit PF1, then **12;2**. The IMS initial entry point to the tutorial is in the **O** for Online option of the primary help menu -- that is why you use **U** for Up from IMS to get to the non-online sections.

To turn off and on such facilities as the COPE Estae, or SYSOUT to IMS log, use DEBUG MODE. The normal mode is DEBUG MODE OFF. DEBUG MODE is completely independent of DEBUG ON/OFF. Only set a DEBUG MODE bit on if you are sure that you need to test in the mode specified by the bit. Some of these bits will intentionally cause abends, under some conditions. The bits, and how to set them on, are described in PF1 then O then 12 then 2 then 2, from ISPF, or PF1 then 12;2;2, from IMS.

DEBUG and DEBUG MODE are intended primarily for use by Compuware personnel, and they can only be completely understood by reading the documentation at the beginning of the ASM source for COPESXP1, UTP1, RC00 and DFSAOUE0. They are mentioned here only to give you some measure of assurance that there are facilities available that help solve problems quickly. The COPE for IMS/DC *Quality Assurance Procedures* manual may also be of interest, if you need more information in this area.

4700 Tracing Support

Support for initial testing of 4700 midname translation is in this release of COPE. It is a trace that comes out on the control region joblog. To turn it on, you:

```
/FOR NODE node
```

Do this from any cleared screen, and name the node that you want to trace. It supports tracing of only one node at a time. If COPE can find the midname in the 4700 FMH, the trace will show (amongst other things, such as RPL fields, TP buffer, etc):

```
MIDNAME=midname
```

Please see the documentation at the beginning of the COPEAOUE member of ASM for further information on this trace.

Sort In-storage Option (COPESORT)

Jobs generated by options 4.1.4 (define groups), 4.2.GS (IMSGEN) and 4.7 (PSBGEN), do large sorts, normally using your system sort program. In the rare case that your system sort program fails, you can change to an in-storage sort (called COPESORT) by coding //COPESORT DD DUMMY in the JCL of the failing job, and coding REGION=32M on the EXEC, or on the JOB if there is already a REGION= on the JOB, and re-running. The presence of this DD (and it must be DUMMY) tells Cope to call COPESORT, otherwise COPE calls regular SORT. To turn this option on permanently, set system variable XCOPEOR to YES in your ZDEFAULT copy member.

The COPESORT in-storage sort will not fail. However, it will use large amounts of above-the-line virtual storage, and this might adversely affect execution time in a high-paging environment. In a low-paging environment, the in-storage sort is slightly (but not significantly) faster than DFSORT and SYNCSORT.

Note that when you code REGION=32M, you are not penalized for "over-specification". MVS does not keep page tables for storage that is not getmained, and COPESORT only getmains the storage it needs. Coding a large REGION= is not a performance consideration, as long as the program you are invoking only getmains the storage it needs, and does not getmain based on region availability. All COPE programs, including COPESORT, are in the former category. You would also want to avoid coding a large REGION= in the case of test programs that could possibly have bugs that cause runaway getmains. COPE programs don't contain such bugs (or haven't in the past).

On some MVS systems, the IBM Assembler (IEV90) has a problem if you code REGION=32M. The problem manifests as assembler steps terminating the job, without an abend or any messages. If you get this problem, you will have to not code REGION=32M on the JOB statement. These jobs do

not normally contain COPE steps that need COPESORT DD DUMMY, but if they do, you will have to code REGION=32M on the EXECs for only those steps.

In summary, the intention of COPESORT DD DUMMY and the associated XCOPEJOR variable is to give you a way around rare problems in system sort programs.

Some detailed notes on the COPESORT routine are:

- To get information on the sorts that run, code a COPESOR1 DD DUMMY as well as COPESORT DD DUMMY. Timings, record counts, and getmains will then appear on SYSTSPRT. Example messages are given in the prolog in the source for COPESORT, in the ASM library.
- If you get a U1991 abend, look on the joblog. You will see a getmain failure message, showing the amount of storage currently gotten. Increase the REGION= and re-run.

SYNC Command Improvements

You issue SYNC to cause Start/Stop status of databases and transactions to be re synchronized, between IMS and COPE.

For example, if a logical system is stopped to COPE, but the IMS status of the databases is not stopped, IMS will still try to allocate the databases. If they are actually unavailable, this will lead to time-consuming allocation failures. Each allocation failure will kick off a COPE transaction that will stop the database. Thus, re synchronization is automatic, but may take many minutes, and adversely affect response time during those minutes. This example will occur on an IMS Release 2 or 3 system after a cold start. On a release 1.3 system, this particular problem does not occur, but a similar situation with started databases being unavailable can occur.

To alleviate this problem, you should do a SYNC NODBR after a Cold Start. You can have this done automatically for you, by coding SYNC NODBR on the COLDS command in the COPEO member of MENUS, and this is the way the COPEO member is delivered.

With COPE Release 3.7.1, there are two changes to SYNC that make this possible:

1. The NODBR parameter specifies that SYNC should issue /STO DB rather than /DBR. This is advantageous after a Cold Start, because there is less overhead than doing /DBR, and /DBR is unnecessary because there are no databases allocated.
2. SYNC goes "both ways" between IMS and COPE status. Formerly, SYNC only did a /DIS STATUS DB, and transferred IMS stopped status to COPE stopped status. It now also reads the COPE Start/Stop records from USTDLMGR, and transfers COPE stopped status the other way, to IMS.

In normal running (other than after a Cold Start), you should not need to issue SYNC. However, if databases or transactions get out of sync, there are three problems that will be fixed by issuing a SYNC:

1. A flood of database allocation failures. Cure by issuing SYNC NODBR. This problem occurs if a set of database datasets become unavailable. For example, if a user or archive utility removes the datasets from disk, and the databases were not stopped in SS (on the DB side) first, IMS will try and allocate them. If you want to make database datasets unavailable, you should P them in SS, on the DB side, first. If you fail to do this, then the cure is to issue SYNC NODBR.
2. Databases are allocated to the COPE system that you want to make available in batch, yet the COPE SS status shows them as stopped. This problem occurs if you fail to issue SYNC NODBR after a cold start. The cure is to issue SYNC (without the NODBR). This is the slower form of SYNC. You omit the NODBR, because you want IMS to actually release the databases, which requires deallocation, not just stopping.
3. Terminals hang. The cure is to issue SYNC NODBR. In a release 1.3 system, database allocation problems, such as the ones described above, will cause terminals to hang. In an IMS Release 2 or 3 system, database allocation problems don't actually cause terminals to hang, but they may increase response time to the point where terminals don't respond for a long time.

In all IMS releases, another reason that terminals may hang is a program somehow becoming stopped. This can happen if you erroneously issue a /STO PROG command, (you should use COPE SS P TR instead), or in some rare message region abends (COPE normally starts the TRAN and the PROG after a message region abend). In all IMS releases, SYNC NODBR will start transactions and programs that are stopped, releasing the terminals.

In IMS release 1.3 systems, the NODBR refinement is not implemented, and you will get a message saying NODBR ignored. In an IMS release 1.3 system, just plain SYNC applies in all cases in the discussion above.

If you get a terminal hang problem that is not cured by SYNC, then you must investigate and increase your PSB, DMB and PSBW pools. Typical large COPE systems require large (2M) PSB and DMB pools in the private area (not in CSA) in the DLS region. The PSBW pool (which is in CSA) must be 300 to 400K, typically. Of course, there are many other, hopefully rare, but possible, reasons for IMS systems to hang, such as archive job failures, PROCOPT=E PSBs, no message regions, etc, and you investigate all these reasons in exactly the same way as for a non-COPE IMS system (/DIS A, /DIS OLDS, /DIS MODIFY ALL, /DIS POOL ALL, DC Monitor, etc). For tuning recommendations, please see the *COPE for IMS-DC Administration Guide*.

Enhanced DYNALLOC and Stage 1 Processing (Common Specs.)

You may now define Logical Systems as having identical Stage 1 and DYNALLOC specifications. This means that if changes are made to any of the DYNALLOC or Stage 1 specifications, the changes will be automatically applied to all the related systems. This approach saves time when there are several identical systems defined for a COPE physical system.

In addition, there is now the capability to edit all the DYNALLOC specifications for all related systems in a single edit session. This allows mass changes of dataset names to be performed quickly.

Adding Stage1 Statements - @COPY

COPE does not look at the terminal network information supplied for the Stage 1 definition, but passes it through to the final Stage 1 output. To avoid the overhead of storing these statements inside COPE, you can put an @COPY <member-name> statement in place of them, that refers to a PDS member that contains the terminal definitions. This statement will be converted to a COPY <member-name> statement in the COPE generated Stage 1 source. Using this facility will save considerable DASD ISPF table space and will also speed up the parsing and regeneration of Stage 1 source members.

The @COPY facility can also be used to add other statements to the Stage 1 source deck. The statements could be special macros that an installation uses to generate IMS statements or could refer to members that contain DATABASE and/or APPLCTN statements for objects outside COPE control.

Xpediter - COPE

You enter Xpediter under COPE option 7.3. COPE enhances Xpediter testing by stopping the interference between multiple users that often occurs with regular Xpediter.

Installing Xpediter support is described in the COPE for IMS/DC *Administration Guide*, and using it is described in the COPE for IMS/DC *Programmers Guide*.

Log Translate for MICS/IMF - COPEARCM

If you are running MICS and/or IMF, or any similar performance reporting package that reads IMS logs, and you wish to have all C-number (COPE) names translated to real names, use the Archive Exit COPEARCM.

Please consider carefully whether you really need this facility before using it. The arguments are:

Advantages

- Real psb names in performance reports/enquiries
- Real trancodes in all cases
- Real database names

Disadvantages

- You will have to keep translated copies of logs, on disk or tape
- You might need to specify options to COPEARCM to control the translation
- For database-specific information, the reports might not be able to distinguish correctly between the same-named database in different systems
- Your users see real trancodes without this support, unless the trancode is an overflow, which is rare
- Programmers can use the Call Trace to see performance problems, rather than MICS or IMF output. The Call Trace is often better suited to their purposes, and is all translated.

- Performance specialists rarely need to translate C-number names, because they are looking for exception conditions based on criteria that don't involve the name. In the cases that they do need to translate, there are many convenient COPE facilities already available.
- If your performance reports feed into other programs, it might be more manageable to use COPEARCM or COPEXLAT to translate them at that point, rather than at the archive point.
- For accounting information, another archive exit, COPEARCX, provides a translated and easy-to-use accounting file.

COPEARCM does the best possible job of translating the log records, and has comprehensive record selection and control options. It will translate any VB or FB file, not just IMS logs.

COPEARCM is described fully at the beginning of its source, in ASM(COPEARCM).

Zdefault Parameter Edit - EVARS

From the COPE primary menu, say EVARS, to go directly into Edit on your Zdefault parameters. This is a convenient way to view and change these variables, particularly useful when you are installing COPE, or a new release of COPE.

Support of BMC DELTA/IMS

From the COPE primary menu, say DELTA, to allow a display of a panel for entry of the Logical System that additions are to be made to. After completion of the DELTA controlled specifications, COPE will update its records in foreground or submit a job to background depending on the setting of option on the panel.

An alternative way to invoke DELTA/IMS is present on the option 4.2.E panel.

COPE translates the PSB names and DBD names to the COPE names via a standard exit present in DELTA/IMS version 4.2. A record of the original input is kept in the USER2 field of the DELTA log.

The COPE DELTA/IMS support is an additional chargeable feature to the base COPE product.

Support of RECON extract for DYNALLOC

Option 4.2.I allows importing of database allocation information stored in the DBRC RECON datasets. COPE maintains the database dataset allocation information in the form of DYNALLOC statements. The 4.2.GD option generates DYNALLOC statements, however the DYNJCL job generated need not be executed to put the generated statements into the authorized library. Option 5.10.14 allows DBRC statements to be generated that fulfill the requirements of DBRC allocation if that option is required.

Support of DEDB Area Datasets

Option 4.2.I or 4.2.E allows importing or editing of DBRC INIT.ADS statements that define DEDB Area Datasets. This information is regenerated with appropriate name changes using option 5.10.14 to update the COPE RECON information.

Support of RACF for ISPF

The Installation manual defines the RACF information to limit access to options under the ISPF portion of COPE. A **Not Authorized** message is displayed for users who try to access options they are not allowed to.

Compress/Expand PDS Utility (Option 3.8)

From the COPE primary menu, say DA, or enter Option 3.8. The utility provides options to review COPE controlled Libraries or to review any PDS dataset. When the COPE name or dataset name

is entered, a summary of the allocated and used space and directory blocks is given. An option exists to generated compress JCL (after making a backup) or to change any of the parameters and generate a job that will expand or contract the dataset size and directory allocation.

Product/Feature Support

COPE provides special processing to support the following products or IMS features.

Table 1: Special Product Feature Support

Product	COPE Release	Description
DB2	2.0	Shared DB2 system
DBRC	2.0	Generated registration
Fast Path	2.2	DEDBs, MSDBs calls/utilities
PL/I	2.2	Param list, IO area lengths
Hogan	3.7.6	Psbname in DFSPRPX0
Xpediter	3.7.6	Multiple user enhancement
OPSMVS	3.7.8	Intercepts /cmds in ctl region
AIB	3.7.8	New IMS AIB call translation
Cobol II	3.7.9	RES/NORES compatibility
APX	3.7.9	Gets DIRCA address
MSC	3.8.2	Remote logon
SLUP MFS	3.8 (proj)	MFS name translation

The following products are known to work with COPE, without any special support being necessary from COPE or from the product.

Table 2: Products not needing special COPE support

Product	Notes
ACF2	--
ASAP	Use Dictionary translate
BMC Delta/IMS	--
BMC VT	Use non-COPE DDBLIB, or translate facilities
BMC 3270	Run DFSERA10 under COPEXLAT listing translator
DBT	--
DC Monitor	-
DFSORT	-
DDS	Use COPEARCM log translate, if desired
DXT	Use non-COPE Dbdlib
HSSR	--
IMF	-
IMS Expert	Use IMS log translate, if desired
IMS/ESA-DB	-
IMS/ESA-TM	-
Msg Requeuer	--
MICS	
Omegamon	
RACF	
Syncsort	
Top Secret	

Products that are clearly un-related to IMS and COPE are not included in the above list (e.g. DASD managers, tape management products, job schedulers, etc).

For batch utilities, you usually have a choice of how to run, either:

1. Use "T" command to translate input control cards and COPEXLAT to translate output listings
2. Run using your non-COPE Dbdlib

Translation facilities are used to support many products and in-house utilities, they are:

- JCL interface to automatically do PSBGENS, DBDGENS, ACBGENS, MFSGENS
- JCL converter for BMP, Batch and Fast Path JCL
- COPE Dictionary (option 6)
- T command, for input control cards
- COPEXLAT, for output listing datasets
- All IMS online commands are translated in and out, e.g. /DIS DB
- COPEARCM Archive Exit IMS log record translation

Index

Symbols

&-variables.....	15
@COPY facility	35

A

Abend	
IEV90	33
in message region.....	31
SOC4 in message region	24
Above Line	
COPE modules	9
ACB	
Generation change for STUBX copy.....	9
ADS Support for DEDB's.....	38
Allocation failures	
Database.....	34
AOI exit	21
Archive log translate.....	36
Backup/Restore	14

B

BMP	
JCL.....	27

C

Call trace.....	29
BMP, DL/1 and DBB	29
Cleanup	
by COPE in message region	25
Cobol	18
and Preload	25
ENDJOB modules	25
Ready Trace	18
Cold start.....	34
Common DYNALLOC specifications	35
Common Stage 1 specifications	35
Comparison of datasets.....	19
Concatenated message region DDnames	10
COPE Menu	
Scrolling	10

COPE Modules

Above line.....	9
COPEARCM.....	36
COPEARCX module	14
COPEIMS1 PSB	16
COPEIMS3 program	16
COPEO panel	15
COPEPGM DD	21
COPE SORT	32
COPEXP2 delete messages	25
COPEXUX exit	8
COPETRAC DD.....	29
CSA usage.....	21
CTL Region JCL	21
Ctlblks IMSGEN	19
Customizing IMS commands	15

D

Database	
Allocation failures.....	34
Datasets	
in Message Regions	26
DB2.....	9, 21
Message Region JCL.....	24
Package Support.....	9
DBB	
Call trace.....	29
Regions	27
DBRC	13,21
Dynalloc extract.....	38
generated statements	13
Region JCL.....	21
Support.....	13
DBRC forward recovery.....	29
DDnames in message region.....	10
DEBUG MODE	31
DEBUG ON	25
DEBUG REGION command	25
DEDB	
ADS Support	38
DELTA/IMS	
Support.....	38
DFSAOUE0 exit.....	21

DFSMPLxx member	22
DL/I	
Batch JCL	18
Call Trace	18
DLISAS or DLI Region	21
DLS online JCL	21
Dynalloc	13
DBRC	38
Extract	38
RECON	38
Source	13
DYNALLOC	38
Mass changes	35
Specification enhancements	35
Dynalloc extract	38
Dynamic Allocation	18
E	
ENDJOB Cobol modules	25
External ACBGEN interface	19
F	
Fast Path	
Message region JCL	19
FASTGEN Feature	8
G	
Getmains in message region	25
H	
HANDLENA specification	
ACBGEN	16
PSBGEN	16
STUBX module	16
Hangs, terminal	34

I	
IEV90 Region=problem	33
Import Option	
Dataset Comparison	19
IMS	

Version 5	9
IMS commands	14
Add or Delete	14
Backup/restore submit jobs	15
customizing	15
LOGOFF	14
Logon en-masse	14
IMS Commands	26, 31, 33
DEBUG MODE	31
DEBUG ON	31
DEBUG REGION	25
HELP (PFK1)	31
REFRESH Bring in changes	26
SYNC	33
TRACE ON LOGIC	31
VERSION LLE display preload	24
IMSGEN	19
ISPF Commands	17
HELP (PFK1)	31
ISPF options	13, 17
4.1.5 Message region datasets	26
4.9 JCL	13
ISPF Options	23, 26, 28, 32
2.5 Program I/O data	17
2.7 call trace	18
3.2 DA alloc libsets	17
3.7 compare and import	18
4.1.4 Libset Definition	14
4.1.4 Sort Failures	32
4.2 IMSGEN	32
4.2.GS IMSGEN	32
4.2.GS Sort failures	32
4.7 Psbgen	17
4.7 Sort failures	32
4.8 Acbgen	17
4.9 JCL	13, 18
4.9 T edit command	13
5.10.14 DBRC	13
5.10.15 Preload	24
5.7 Import	19
7.2 call trace	18
JOB command	17
ISPFOptions	28

J	
JCL	21, 27
BMP, DL/1 and DBB	27
Conversion	19
CTL, DLS, DBRC	21

Message region	22
JOB	
Check/resubmit jobs.....	16
JOB command	16
L	
Libset allocation	17
MULTIPLY command.....	17
PRI command	17
SEC command.....	17
VOL command	17
Log Translate	36
Logical System	
Specification	18
LOGOFF Command.....	14
Logon User	14
LSO=S in CTL region.....	21
M	
MAC type	14
Menu	
Scrolling	10
Message region	
Cleanup by COPE	10
Concatenated DDnames.....	10
DDnames	10
Getmains	25
JCL.....	19
Program loading	10
STEPLIB	10
Message Region.....	19
Datasets.....	26
DB2 JCL	24
Fast-Path JCL.....	19
Message Region JCL.....	22
MQM	
Support.....	8
MSC	
Synchronized MFS	8
MULTIPLY command.....	17
N	
Nucleus IMSGEN.....	19

P	
PDS	38
Compress utility.....	38
Expand utility	38
PFK1 (Help).....	31
Preload	
and Cobol	25
in Message region	24
PRI command	17
Primary Option Menu	17
Program loading in message regions	10
PSBGEN.....	17
R	
RACF	
Support.....	38
Ready Trace (Cobol).....	18
RECON	
Dynaloc extract.....	38
REFRESH brings in changes.....	33
REGION=assembler problem	33
S	
S0C4 in message region	24
SEC command	17
Sort	
Abends.....	32
In-storage	32
SQL	
Call trace.....	18
SSM member	21
Stage1	35
@COPY statement	35
common definitions	35
specification enhancements.....	35
STEPLIB	
in Message region	10
In MSG region.....	23
STUBX	
ACB generation change	17
STUBX module	17
SUB command	15
SYNC command.....	33
SYSUDUMP DD	23

T		ViaSoft	
Terminal Hangs.....	34	Support.....	10
TRACE ON LOGIC.....	31	Virtual Date feature	8
Traces		VOL command	17
DL/1 call trace.....	18		
Ready trace (Cobol)	18	X	
SQL call trace	18	XCOPEPGM parameter.....	21
Translate JCL file names.....	13	XCOPEPK parameter	9
TRIMAR		XCOPESOR variable	32
Secondary index utility	9	Xpediter.....	36
U		Z	
USTDLMGR	14	Zdefault	
		EVARS	37
V		Variable edit, EVARS.....	37
VERSION LLE command	24		