IBM Application Program GH20-0788-2

CALL – OS

Operator's Manual

Program Number 360A-CX-42X

This publication describes the operating procedures for the CALL-OS system, which is a terminal oriented, time sharing system. CALL-OS is designed to function under the control of an IBM System/360 Operating System in one of two configurations: Multiprogramming with a Fixed Number of Tasks (MFT) or Multiprogramming with a Variable Number of Tasks (MVT). With either configuration, the installation may use the CALL-OS Batch Interface facility, which permits the terminal user to submit jobs to OS/360 batch processing.

This publication is intended for computer center personnel responsible for the day-to-day operation of the system. Topics include the procedures for starting the CALL-OS system, techniques required during system operation, and the procedures for shutting the system down. The operator communicates with the system through the operator command language, which is discussed in detail, along with the messages and codes issued by the system.

Note: The CALL/360-OS system has been renamed the CALL-OS system. Thus, documentation of Version 2 of the CALL/360-OS system refers to the system as CALL-OS.

Terminal Equivalency

Terminals which are equivalent to those explicitly supported may also function satisfactorily. The customer is responsible for establishing equivalency. IBM assumes no responsibility for the impact that any changes to the IBM supplied products or programs may have on such terminals.

Third Edition (March 1972)

This is a major revision of, and obsoletes, GH20-0788-1. Additions and corrections have been made throughout the manual, which should be reviewed in its entirety.

This edition reflects Version 2, Modification Level 0 of the CALL-OS time sharing system and all subsequent versions and modifications until otherwise indicated in new editions or Technical Newsletters. Changes are continually being made to the specifications contained herein. Therefore, before using this publication, consult the latest System/360 SRL Newsletter (GN20-0360) for the editions that are applicable and current.

Copies of this and other IBM publications can be obtained through IBM branch offices.

A form has been provided at the back of this publication for reader comments. If this form has been removed, address comments to IBM, Technical Publications Department, 1133 Westchester Avenue, White Plains, N. Y. 10604.

PREFACE

This manual describes the operating procedures and system messages of the CALL-OS time sharing system. The manual contains seven major sections. The introduction describes the CALL-OS system and its relationship to the operator. The second section describes online operations in detail: topics include startup procedures, operating techniques, and shutdown and restart procedures. A separate section provides, in alphabetic sequence, a listing and explanation of the operator command language. The last four sections contain all system messages and abnormal termination codes, organized according to the device on which they appear. Additional material is contained in the appendices.

The operator is assumed to be a trained and experienced OS/360 operator with a working knowledge of the following associated publications:

IBM System/360 Operating System:

Operator's Procedures, GC28-6692

Operator's Reference, GC28-6691

Messages and Codes, GC28-6631

The operator should also be familiar with the following publications which describe the CALL-OS system in general and the terminals used:

CALL-OS System Description Manual, GH20-0673

CALL-OS Terminal Operations Manual, GH20-0787

Additional information on the CALL-OS system and its utilities is found in CALL-OS Executive and Utilities Program Description Manual, GH20-0786; the material in this manual should be used at the direction of the installation system programmer.

CONTENTS

TABLES

INTRODUCTION

CALL-OS provides a computing service to the user at a remote terminal, while running concurrently with OS/360 background batch work. By means of time sharing, each individual user at a remote terminal feels that he is the sole user of the system. CALL-OS operates as a standard job under the IBM System/360 Operating System, Multiprogramming with a Fixed Number of Tasks (MFT) or Multiprogramming with a Variable Number of Tasks (MVT) -- hereafter referred to as OS/360. CALL-OS occupies a single partition or region, respectively, hereafter referred to as a task area.

The rest of this section summarizes the CALL-OS system, its operating requirements, and the messages it produces.

ORGANIZATION OF THE SYSTEM

CALL-OS can be visualized as consisting of five major parts: the control program, the processing programs, the command languages, the data base, and the CALL-OS Batch Interface (COBI) facility. Each of these is described briefly as it affects the operator.

CONTROL PROGRAM

The control program (also called the executive) schedules and dispatches all operations within the system, and provides terminal and disk input/output. Once the system is running, the operator may receive messages from the executive pertaining to system status. He may also request status information and, in some ways, control system processing.

PROCESSING PROGRAMS

The processing programs include three compilers and the utilities. The compilers for CALL-OS make the BASIC, FORTRAN, and PL/I langauges available to the terminal user. These languages are described in the appropriate reference manuals and, except for system errors detected during compiler operation, the operator should not be affected by their use. The utilities include system build procedures, initialization routines, and programs which manipulate the CALL-OS data base and the COBI data sets. Their use is directed by the installation system programmer and he should be notified when messages pertaining to these utilities appear. No further operator action is required.

COMMAND LANGUAGES

CALL-OS provides two command languages, one for the operator and one for the terminal user. The operator command language allows the computer operator to initiate processing, control the number of lines and users, request system and user status, and perform other online functions. The terminal command language allows terminal users to enter and execute programs, share programs and data files, store and execute object programs, submit jobs to OS/360, and perform other functions from his terminal. The computer operator may use both languages. The terminal user may not use the operator command language. The operator command language is described in a subsequent section of this manual; the terminal command language is described in the publication CALL-OS Terminal Operations Manual.

DATA BASE

The data base is a collection of data sets, used for the storage and retrieval of system and user resources. It consists of three logical parts:

- The system base, which fills system requirements for storage areas
- The user base, which fills user requirements for storage areas
- The index, which identifies all the data sets associated with the system and user bases

The system base contains system-oriented information; it consists of those data sets needed for compiler and overlay module storage. In addition, data sets are required for work areas.

The user base contains user-oriented information, whether the user is the operator at the command console or a remote user at a terminal. In either case, the user base contains programs and data files associated with the user and the information necessary to store and retrieve these files. The user base has two parts: the system group and the user group. Program and data files entered by the operator are stored in the system group and are available to all users. Program and data files entered by terminal users are stored in the user group and, if the user desires, may be made available to other users.

CALL-OS BATCH INTERFACE (COBI)

COBI is an optional feature of CALL-OS. When COBI is used, terminal users can prepare OS/360 jobs and submit these jobs directly from the terminal to be processed in the OS/360 batch processing environment. All or a portion of the job output may be saved for later printing at the terminal. Depending on an initialization parameter, the operator may have to initiate reading of these jobs into the system. This is done in the same way he initiates the reading of jobs under MFT or MVT. In addition, he may be required to perform other actions which aid in the successful operation of the system and the execution of user-submitted jobs.

REQUIREMENTS FOR OPERATING CALL-OS

Use of the CALL-OS system involves two types of operations: online operations, which take place while CALL-OS is running, and offline operations, which take place while CALL-OS is not running. The way in which these requirements are communicated to the operator and the types of things he may be required to do are described in the following text.

ONLINE REQUIREMENTS

CALL-OS provides for interaction between the operator and the online system through the operator command language and console messages. The consoles involved are the OS/360 system operator's console, which must be available when OS/360 is in use, and the CALL-OS consoles, described in the following paragraph.

CALL-OS uses two types of consoles for operator/system communication: the command console, from which the operator issues the commands necessary to initialize, run, and shut down the online system, and the communications console, at which he receives system status and diagnostic messages. Any CALL-OS system must have at least one command console (two may be used) and a communications console. However, if a communications console is not available, the messages intended for it appear on the OS/360 system operator's console.

The consoles are assigned to specific logical line numbers with parameters in the EXEC statement of the startup deck (see Appendix C); these logical line numbers are associated with specific terminal types by the DD statements included in the startup deck. Any CALL-OS console may be either a Teletype Unit* (Type 33 or 35) or an IBM 2741 Communications Terminal (Type Correspondence or EBCD). If IBM 2741 Communications Terminals are used, the correct print element must be used on the terminal to ensure readable output: the print element for the Correspondence terminal is 1167087; the print element for the EBCD terminal is 1167643.

Since the operator for CALL-OS is concerned primarily with the online operations, these operations and the commands used to perform them are described in separate sections.

OFFLINE REQUIREMENTS

The offline operations for CALL-OS involve the execution of the utility programs, and, if COBI is used, the starting of a special COBI writer program under certain conditions. Messages pertaining to offline processing appear on the OS/360 system operator's console and the system printer. The operator intervention required for offline operations is minimal, and is described in the following text.

Executing the Utility Programs

The utilities execute as standard jobs, in a separate task area, under control of OS/360. Because their use is directed by the installation system programmer, the requirements for executing them and the options available are described in detail in the publication CALL-OS Executive and Utilities Program Description Manual. The operator must monitor the messages and notify the responsible system programmer when an error occurs. Under no circumstances should a utility be restarted without consultation with the programmer.

Starting a COBI Writer

If CALL-OS is no longer operating but jobs submitted through COBI must either still be run or have their output processed, the operator must start the COBI writer program. This program intercepts JCL for later scanning at the terminal and makes SYSOUT data sets not saved for scanning available to the OS/360 output writer. While CALL-OS is operating, the DIBWTR functions are handled by the module M#JCL.

The operator starts the COBI writer by issuing the following command at the OS/360 system operator's console:

```
S DIBWTR
```

When all of the output for the COBI output class has been processed, the operator may terminate the writer by issuing the following command at the OS/360 system operator's console:

```
C DIBWTR
```

Note: The operator does not receive a message instructing him to start the COBI writer. If the writer is not started, the COBI message class associated with the most recent CALL-OS session is not processed; the next time CALL-OS is initialized with COBI, this output class is processed in the normal way.

* Trademark of the TELETYPE Corporation

SYSTEM MESSAGES

Messages from the CALL-OS online system and the offline utility programs may appear on one of four possible devices, as follows:

- OS/360 system operator's console, which receives CALL-OS initialization messages, online ABEND messages, COBI operating messages during online and/or offline operation; and utility ABEND codes and messages during offline operation.

- CALL-OS command console, which receives messages in response to operator commands issued at the console during online operation

- CALL-OS communications console, which only receives messages during online operation; these messages include system messages from the executive, user-related error codes, disk I/O error messages, and compiler-related messages

- OS/360 system printer, which receives messages from the CALL-OS offline utilities

Subsequent sections list the messages which appear at each device. Within each group or subgroup of messages, the messages are listed in alphabetic or numeric order, whichever is appropriate. An explanation of the cause of each message is also given.

ONLINE OPERATIONS

This section describes the procedures required to get the system up and running, the actions which may be required while the system is in operation, and the procedures required to terminate system operation. The major facility used in all these activities is the operator command languge, which is described in the next section.

INITIALIZING THE SYSTEM

This subsection is written for the operator who is familiar with normal operating procedures of the OS/360 system generated for use with CALL-OS. The CALL-OS system is started via normal OS/360 job management facilities with certain assumptions made as to the structure of the version of OS/360 to be used. For example:

1. CALL-OS is usually the highest priority job.
2. In order to facilitate initialization, a unique job class is assigned to the task area in which CALL-OS is to be executed. This class is specified on the JOB statement in the startup deck and an initiator is started for it.
3. The systems programmer or analyst has selected the task area size and a list of resident modules that are optimal for this particular system.
4. All requisite JCL options have been selected and are included in the CALL-OS startup deck or procedure library.

CALL-OS initialization routines are invoked by use of a startup deck or a comparable procedure, either of which is supplied by the installation system programmer. (See Appendix C for a sample startup deck and a summary of the initialization options and the JCL statements in the deck.) Various warning messages may be printed at the OS/360 system operator's console with the final notice of a successful initialization being:

```
DIBIN001 CALL-OS INITIALIZED. NJA=nnnnnn,OJA=mmmmmm
```

The operator can now establish command console communication with the system and enable the system's telephone lines for operation.

ESTABLISHING CONSOLE COMMUNICATION

To begin work with the CALL-OS system, the operator must first ensure that the teleprocessing control units are set up properly. He must then establish communication through one or more designated consoles. Regardless of the type of console or function involved, a console is made operative in two phases:

- Dialing in to establish the connection with the system
- Identifying the console to the system

Table 1 gives the procedures for dialing in for both the command console and the communications console. The procedure used depends on the terminal type and whether the terminal has a dial-up line or a private line with a ring-down circuit.

Table 2 gives the procedures for identifying a command console to the system. The procedure used depends on the terminal type.

Table 3 gives the procedures for identifying the communications console to the system. The procedure used depends on the terminal type.

Table 1. Procedures for dialing into the system

Type of Line	IBM 2741 Communications Terminal	Teletype Unit, Type 33/35, Model ASR/KSR
Dial-up	1. Set ON/OFF switch (at right of keyboard) to ON. 2. Set COM/LCL switch (Communicate/Local, in niche in left side panel) to COM. 3. Press TALK button on data set. 4. Lift data set receiver, wait for dial tone and dial system telephone number. Response: steady high-pitched tone. (See note.) 5. Firmly press DATA button on data set. 6. Put receiver back in cradle. Connection has been made. If DATA button light goes out, go to Step 3 to remake connection. 7. See Tables 2 and 3.	1. Set volume control for audible tone. 2. Press ORIG button (Originate, at left end of row of buttons below telephone dial) and wait for dial tone. 3. Dial system phone number. Response: steady high-pitched tone. If no answer or line is busy, disconnect (press CLR and repeat from Step 1. Connection has been made. 4. See Tables 2 and 3.
Private line with ring-down circuit	1. Hold Ring button (xxxx) down three seconds (located on 103A data set).	1. Set telephone ALT MODE key to LOCAL (or equivalent non-DIAL position). 2. Press ORIG button. 3. Press Ring button for three seconds.

<u>Note</u>: If there is no answer, the system has not been properly initialized. If the line is busy and the console is the command console, cancel CALL-OS; a busy signal indicates that the command console has been initialized by an unauthorized user; if the console is the communications console, disable the line and dial in again.

Table 2. Procedures for identifying the command console to the system

IBM 2741 Communications Terminal (Correspondence or EBCD)	Teletype Unit, Type 33/35, Model ASR/KSR
1. Type the characters LOG (or LOGON) and depress RETURN, next to ON/OFF switch. See note 1. 2. Wait for system to print greeting: USER NUMBER,PASSWORD-- 3. On same line, type SYSLIB. Depress RETURN. See note 2. 4. System's sign-on invitation plus operator's typed response would appear as: USER NUMBER,PASSWORD--SYSLIB READY After identification is accepted, system types READY. Operator can then continue his startup procedure.	1. Wait for system to print greeting: USER NUMBER,PASSWORD-- (Type 35 ASR only.) Press K key (on left control panel) to unlock keyboard. 2. On the same line, type SYSLIB. Depress RETURN. See note 2. 3. System's invitation plus operator's response would appear as: USER NUMBER,PASSWORD--SYSLIB READY After identification is accepted, system types READY. Operator can then continue his startup procedure.
Notes: 1. This ensures that the correct translation code is used for the terminal type. If any characters other than LOG (or LOGON) are entered and/or RETURN is depressed, the system assumes that the terminal type is the one defined in the startup deck for the logical line number assigned to the command console. The translation code for that terminal type is used to print the greeting message in Step 2. 2. An actual user number and password would be used infrequently and only when the operator wishes to test the system's response to an individual remote user's number and password and/or to run such user's program.	

Table 3. Procedures for identifying the communications console to the system

IBM 2741 Communications Terminal (Correspondence or EBCD)	Teletype Unit, Type 33/35, Model ASR/KSR
1. Type the characters LOG (or LOGON) and depress RETURN key (see note). System acknowledges with: AT tt:tt LINE nn COMMUNICATIONS CONSOLE IS ACTIVE. SYSTEM ENABLED AT hh:mm If RETURN only is depressed, the system assumes that the terminal type is the one defined in the start-up deck for the logical line number assigned to the communications console; the system accordingly prints the preceding message in that code. Thereafter, the communications console keyboard is locked. System messages only will be received.	1. RETURN key need not be pressed. Response: AT tt:tt, LINE nn COMMUNICATIONS CONSOLE IS ACTIVE. SYSTEM ENABLED AT hh:mm. Keyboard is not locked. Unit serves only to receive system error messages.

Note: If any characters other than LOG (or LOGON) are entered and RETURN is depressed, the system responds with a two line message. One line will be readable data and the other will be garbled data; the terminal type determines whether the readable data is the first or second line, as follows:

Correspondence	EBCD
TYPE LOG OR RETURN KEY	garbled data
garbled data	TYPE LOG OR RETURN KEY

The operator must repeat Step 1 until the acknowledgement message is printed.

ENABLING THE SYSTEM

After identifying himself to the system, the operator should create the user sign-on message using the *DATE command and, if applicable, the *MESSAGE command. Following the entry of these two commands, the operator must issue the *ENABLE command to allow users to dial in to the system. These three commands are described in detail in the section "Operator Command Language".

The following example shows the complete sequence of commands and responses used to enable CALL-OS:

```
Operator:   *DATE
System:     ENTER
Operator:   10/29/71    FRIDAY
System:     READY
Operator:   *MESSAGE
System:     ENTER
Operator:   CALL-OS UP UNTIL 1100, RUN ***NEWS FOR NEW HOURS
System:     READY
Operator:   *ENABLE
System:     READY
```

The READY messages indicate that the requested function has been completed successfully. For example, the final READY message indicates that all lines have been enabled.

When a user dials in, this system would respond with the following sign-on message:

```
ON AT 9:20      10/29/71 FRIDAY LINE 14
CALL-OS UP UNTIL 1100, RUN ***NEWS FOR NEW HOURS
USER NUMBER, PASSWORD--
```

In this example, the information 10/29/71 FRIDAY was the operator's response to the *DATE command. The second line was the response to the *MESSAGE command; the program ***NEWS is an installation-written program which supplies new hours for system operation. Note that the underlined portions of the sign-on message cannot be changed by the operator.

USE OF THE OS/360 SET COMMAND

The online accounting procedures incorporated in CALL-OS depend on the integrity of the OS/360 clock times being maintained throughout the duration of the run. Therefore, the SET command should not be issued from the OS/360 system operator's console while CALL-OS is running. If the OS/360 time or date is reset, erroneous information is recorded for all CALL-OS users active at the time the SET command was issued.

OPERATING TECHNIQUES

While the CALL-OS system is in operation, the operator may have to perform additional actions. These actions may involve either assisting the terminal user or providing additional services when COBI is in operation. Each of these activities is described in greater detail in the following text.

USER ASSISTANCE BY CENTER OPERATOR

The CALL-OS operator may be required to respond to minor complaints from the terminal user. Problems may be classified as human, hardware, and software. It is expected that human errors can be resolved between operator and user. It is the responsibility of the systems programmer or analyst to investigate and define hardware and software problems. The operator can assist in those areas by making use of the system commands *STATUS, *REPORT, etc. to further the investigation.

Loss of Data or Sign-on Problems

It may happen that a terminal user will report that he is experiencing a loss of data or difficulties in signing on to the system. The computer center operator must therefore direct the terminal user to check for the following conditions:

- Is terminal firmly plugged into the power socket?
- Is keyboard power switch positioned to ON?
- Is switch on side of terminal positioned to COM (COMMUNICATE)? (Valid for 2741 only.)
- Is the correct password being used?

Performing Sign-On Procedure

Direct the user of the 2741 terminal to reestablish communication with the computer center as follows:

1. Switch 2741 terminal off.

2. Place the data set in talk mode (by pressing the TALK button) and dial the computer center number.

3. When a high-pitched tone is heard, press the DATA button and replace the receiver on the rack.

4. Switch 2741 terminal on.

5. Type the characters LOG (or LOGON) and depress the RETURN key.

5a. If any characters other than LOG(or LOGON) are entered and/or RETURN is depressed, the system assumes that the 2741 terminal type is the one (correspondence or EBCD) defined in the startup deck for the logical line number assigned to the terminal.

5b. If the sign-on message is garbled in the case of 5a, it means either that the default 2741 terminal type is incorrect or the incorrect print element is being used on the terminal. The user can rectify an incorrect default by carrying out one of the following:

   - Type LOG and depress RETURN.
   - Type a user number and password correctly.

   Otherwise ensure that the correct print element is being used: 1167087 for the correspondence, 1167643 for the EBCD.

For a Teletype unit, Type 33/35, Model ASR/KSR, the steps are:

1. Turn up volume control sufficiently for audible tone; press the ORIG button, and wait for a dial tone.

2. Dial the computer center number. (Response is a steady high-pitched tone.) If there is no answer or the line is busy, disconnect (press CLR) and dial again.

   For Type 35 ASR, press K key (on left control panel) to unlock keyboard.

3. Sign on.

Disabling Telephone Lines

To prevent a user from attempting to establish a connection over an out-of-service line between the telephone company switching station and the computer, the operator needs the ability to prevent (busy out) any particular line from being accessed by the rotary line selector (or an equivalent device used by the telephone company). The manner in which this is done is dependent on the particular type of telephone equipment at this installation. When the operator receives a message to busy out a particular line, he should therefore consult his telephone equipment manuals for details of the specific procedure required. He should also disable that line through the *DISABLE command.

Allocation of Storage

If the amount of core storage available to user programs is limited, the likelihood of running more than one user program at a time is decreased. This means that user terminal response times may suffer during peak usage, resulting in user complaints of service delays. The amount of core available for the execution of user programs is printed on the OS/360 system operator's console after the system is successfully initialized (message DIBIN001). The operator should be aware of the normal new and old job area sizes; he should bring any deviation in these sizes to the attention of the installation system programmer.

COBI OPERATING TECHNIQUES

If COBI is used, the operator may be required to perform additional actions while the system is in operation. These actions include:

- Starting a COBI reader to read submitted jobs into OS/360
- Performing user-requested services

Each of these is described in more detail in the following text.

Starting a COBI Reader

Jobs submitted under CALL-OS with COBI are written into two data sets, known as SYSINA and SYSINB. They are read into OS/360 for execution by the COBI reader cataloged procedures, DIBRDRA or DIBRDRB, respectively. The reading of the data sets may be initiated either automatically by COBI or manually by the operator, depending on the mode of operation in effect at the time a data set becomes available for reading. If the automatic mode is not used and both data sets are full, no more jobs may be submitted to COBI until one of the data sets has been read.

Therefore, during the operation of a CALL-OS system with COBI, the operator may be asked to start a reader if the manual mode of starting a reader is in effect for the session. The operator is instructed when to start a reader by messages which appear on the OS/360 system operator's console. These messages have the following format:

```
    DIBxxxxx  text .... START DIBRDRx
```

where

xxxxx is the message identifier

text is an informative message which indicates the condition causing a reader to be needed

x is either A or B; it indicates which reader procedure is to be started to read the input data set.

The operator must then issue a start reader command on the OS/360 system operator's console. This command is identical to that used for an OS/360 reader except that the operator specifies either DIBRDRA or DIBRDRB. The following example shows a message instructing the operator to start a reader and his response to that message:

```
    System:    DIBEX021  20 JOBS ON SYSINB, START DIBRDRB
    Operator:  S   DIBRDRB
```

The S DIBRDRB command causes all the jobs on the associated input data set to be read into OS/360 for processing by initiators for the appropriate job classes. However, it may be desirable to start the reader at a particular job rather than at the beginning of the data set. To do this, the operator uses the following format of the start command:

```
    S DIBRDRx,,,job-name
```

where

x is either A or B

job-name is the name of the first job to be read

All jobs preceding the specified job-name are skipped, as well as all write-to-operator messages associated with these jobs.

The operator may also use the start command to display the names of the jobs on a COBI input data set. To do this, he uses the following format of the start command:

```
    S DIBRDRx,,,$$$$
```

where

x is either A or B

This command causes the names of the jobs on the associated input data set to be displayed on the OS/360 system operator's console. However, the jobs are not read into OS/360 for processing.

Other operator actions in connection with the COBI reader are identical to those he performs for the OS/360 reader. See the operator's procedures and reference manuals for the IBM System/360 Operating System.

Note: The operator may use the *COBI command to specify which mode of reader operation is to be in effect. (See the description of the RESET function of the *COBI command.)

Performing User-Requested Services

The operator may receive messages which originate from a user's terminal. These messages could contain setup instructions for jobs submitted from the terminal or status information regarding the execution of the job. These messages appear either on the OS/360 system operator's console or on the CALL-OS communications console.

Messages intended for the OS/360 system operator's console are identified by the identifiers DIBEX002 or DIBRD003. Messages intended for the CALL-OS communications console do not have an identifier if they appear on the communications console; if, however, the communications console is not active, these messages appear on the OS/360 system operator's console preceded by the identifier DIBEX001.

SHUTDOWN AND RESTART

The operator uses two commands to shut the system down in an orderly manner. The *WARN command transmits a message to all current users that the system is to be shut down; this message should be issued several minutes before the *OFF command to allow users to terminate their work. The *OFF command causes the CALL-OS job to terminate; all CALL-OS close-out procedures are invoked, the OS/360 environment is restored, and the final return is made to OS/360.

These two commands ensure proper shutdown of the system, whether the shutdown is because the normal operating period is at an end or because of some other reason (for example, an OS/360 malfunction). The job termination message appears on the OS/360 system operator's console. This is because CALL-OS terminates approximately one minute after the *OFF command is issued.

The following example shows the sequence of operator commands to be used to shut down the system:

```
Operator:  *WARN
System:    ENTER
Operator:  SYSTEM WILL BE SHUT DOWN AT 16:00
System:    READY
Operator:  *OFF
System:    SYSTEM OFF AT 16:01
```

Note: In the event of a system failure, there are no built-in restart procedures. The center operator must reinitialize the system as explained in "Initializing the System."

OPERATOR COMMAND LANGUAGE

The operator communicates with CALL-OS by means of the operator command language. These commands may be used only by personnel at the central computer installation. With these commands, the operator may:

- Enter the current date and other information into the user's sign-on message - *DATE
- Transmit a message to all users at sign-on time - *MESSAGE
- Enable all or one or more terminal lines - *ENABLE
- Control the percentage of total processing time given to batch processing - *BATCH
- Validate a new user onto the system - *VALIDATE
- Cancel a user from the system - *CANCEL
- Transmit a message, during system operation, to another terminal or console - *TELL
- Print system status information - *REPORT
- Print user status information - *STATUS
- Print the number of users either on the entire system or on a range of lines - *USERS
- Request that the system disable all lines not in use and prevent new users from signing on - *IGNORE
- Request status of COBI jobs and/or data sets, scratch COBI data sets, or override COBI initialization parameters - *COBI
- Disable one or more terminal lines - *DISABLE
- Transmit a warning message to all users - *WARN
- Terminate system operation - *OFF

A brief discussion of the procedures used for entering commands precedes the command formats and descriptions.

ENTERING COMMANDS

The operator commands may be entered only at the command console and are identified by an asterisk, which precedes the command name. Except where noted, each command may be abbreviated to the asterisk plus the first two characters of the command. For example, *BATCH may be abbreviated to *BA. One or more parameters may be required; their formats are given in the following subsection.

The operator must depress the RETURN key after he enters either the command or the parameters, if any are necessary. If the requested operation is completed successfully, the system responds with READY unless additional information is required. In this case, the system responds with ENTER. The operator then enters the additional information; for example, the date or a message. He then depresses the

RETURN key. The system responds with READY when the requested information has been processed successfully.

If the operator wishes to correct or delete the current line, he may do so by following the procedures given below. The procedure used depends on the terminal type of the command console.

CORRECTION PROCEDURES FOR THE IBM 2741 TERMINALS

The operator may use correction procedures to correct or delete the line he is currently entering on the command console. These procedures apply to both the 2741 Correspondence and the 2741 EBCD terminal.

To Correct a Typing Error

The operator may correct a typing error on the current line as long as he has not depressed the RETURN key for the line. The procedure is:

1. Backspace to the point of the error
2. Press the attention (ATTN) key, which is located above the ON/OFF switch.

The system underlines the character in error and moves the paper up one line without moving the print element. The operator then resumes typing from the point of error.

To Delete the Current Line

The operator may delete the current line as long as he has not depressed the RETURN key for the line. The procedure is:

1. Type the degree (¢) symbol, which is an upshift J
2. Press the ATTN key

The system prints the word DELETED and ends the line automatically.

CORRECTION PROCEDURES FOR THE TELETYPE UNITS

The operator may use correction procedures to correct or delete the line he is currently entering on the command console. These procedures apply to both the Type 33 and the Type 35 Teletype Unit.

To Correct a Typing Error

The operator may correct a typing error in the current line as long as he has not depressed the RETURN key for the line. The procedure is:

1. Hold down the SHIFT key
2. Strike the letter O once for the first incorrect character in the line and once for each character following it (including blanks)

The system prints either a back-arrow symbol or an underline symbol for each upshift O. The operator then types the correct character and all characters following it on the same line.

To Delete the Current Line

The operator may delete the current line as long as he has not depressed the RETURN key for the line. The procedure is:

1. Hold down the control key (CTRL) which is at the left of the third row on the keyboard

2. Strike the letter X

The system prints the word DELETED and ends the line automatically.

OPERATOR COMMAND FORMATS

This subsection contains formats for the operator commands. The following conventions are used in the formats:

- Uppercase letters represent required information and must be entered as shown.

- Special characters (for example, asterisks, commas, hyphens) must also be entered as shown.

- Lowercase letters represent optional information, to be supplied by the operator

- Parameters enclosed in brackets may be omitted.

- An ellipsis (...) indicates that the preceding parameter may be repeated as many times as necessary.

The commands are in alphabetical order. The purpose and use of each command is given as well as one or more examples.

Note: Many command formats require the operator to use a logical line number to refer to a particular terminal. This number reflects the order in which the CALL-OS terminals are assigned. Teletype terminals are assigned first, then 2741 correspondence terminals, and finally 2741 EBCD terminals. If any terminal type is missing, logical line numbering continues consecutively with the next type. For example, if a system has three teletype terminals and three 2741 EBCD terminals, the following logical line numbers are assigned:

Teletype - 1, 2, and 3
2741 EBCD - 4, 5, and 6

If, during another session of CALL-OS, 2741 correspondence terminals are added, the logical line numbers of the 2741 EBCD terminals shift accordingly.

*BATCH COMMAND

The *BATCH command allows the operator to control the amount of CPU time allocated to background processing of lower priority OS/360 jobs. The format of the *BATCH command is:

```
*BATCH x[,yy]
```

where

x determines the mode of allocation and is one of the following values:

0 automatic mode; that is, CALL-OS dynamically adjusts CPU allocation according to the current number of active users. In this case, yy is not required.

1 yy percent of the available old job CPU time is allocated to lower priority tasks.

2 yy percent of the available new job and old job CPU time is allocated to lower priority tasks.

yy is the percentage of time to be allocated to background work and must be within the range of 0 through 95. If zero, only CALL-OS WAIT time is allotted to background work. Any number that is not a multiple of five is reduced to the next lowest multiple of five percent.

Note: This command has a great impact on both the terminal user's response times and local batch throughput; its use should be directed by the installation system programmer or analyst.

Automatic mode, or mode 0, allocates a certain percentage of total old job time to background processing. The percentage allocated is based on the current number of terminal users and is adjusted automatically as users sign on and off the system. This mode affects old jobs only, new jobs are not preempted for batch processing. Mode 0 is in effect immediately after CALL-OS is initialized.

Mode 1 also allocates a percentage of available old job time to background jobs. The percentage is specified in the command and is used as the basis for sharing time with the background, regardless of the number of users on the system.

Mode 2 differs from Mode 1 in that it allocates a specified percentage of available old and new job time to background processing. This mode should be used with extreme caution, and only when long response times can be tolerated by the user.

In practice, many background jobs may require only a small portion of CPU time in which to commence an I/O operation before they are forced to wait for its completion. Therefore, in all three modes, it is desirable to allocate time in small time increments to allow the overall system to operate efficiently. The increment is specified in the SHRTSL option in the startup deck.

It should be noted that an old job (or with Mode 2, a new job) receives its entire time slice before being swapped out. The time, however, is allocated in SHRTSL increments; the total elapsed time in core is a function of the percentage of time to be shared with the background as well as the amount of that time actually used by the background.

The following example shows the use of the *BATCH command to allocate 55 percent of the available old job CPU time to lower priority tasks.

```
Operator:  *BATCH 1,55
System:    READY
```

The operator may restore automatic mode by entering the following command:

```
Operator:  *BATCH 0
System:    READY
```

*CANCEL COMMAND

The *CANCEL command deletes a specified user number from the set of valid user numbers. The equivalency file is searched for the specified user number. If a match is found, the password for that user number is set to zero. If a match is not found, a message is printed at the command console. The format of the *CANCEL command is:

```
*CANCEL aaannn
```

where

aaannn is the user number to be cancelled and consists of three letters aaa and three numbers nnn.

The following example shows the use of the *CANCEL command to cancel user numbers IBM408 and IBM453:

```
Operator:  *CANCEL IBM408
System:    READY
Operator:  *CANCEL IBM453
System:    NO SUCH USER
```

In the first case, the equivalency file is searched for user number IBM408; the READY message indicates that a match was found and the password set to zero. In the second case, the equivalency file is searched for user number IBM453, the error message indicates that a match was not found.

Note: Unless specifically purged, any program or data file left in the catalog for a cancelled user number would be available to a future user assigned to that number. In addition, any entries in the * or ** directories pooled by the cancelled user are not removed. Also, cancelled user numbers and all associated files are copied into a reorganized data base or onto a backup tape by the data base utility. For information on removing a user from the data base, see CALL-OS Executive and Utilities Program Description Manual.

*COBI COMMAND

The operator may use the *COBI command only when the COBI facility is being used in the current session of CALL-OS. The command has several functions, each of which is indicated by a parameter following the *COBI. The functions and the associated parameter are as follows:

- Initiate the reading of any jobs in a COBI input data set into the OS/360 job stream - *COBI ANYBATCH
- Request the status either of the data sets associated with COBI jobs or of a COBI input data set - *COBI DSSTATUS
- Request the status of any jobs submitted through COBI - *COBI JOBSTATUS
- Override specifications made in the COBI startup parameters - *COBI RESET
- Scratch JCL and/or scannable data sets associated with a COBI job - *COBI SCRATCH
- Inhibit or allow the submitting of jobs from a terminal - *COBI SUBMIT

Except where noted, any form of the *COBI command may be abbreviated in a manner identical to the other operator commands. That is, *CO followed by the function, which may also be abbreviated to the first three letters. For example, *COBI ANYBATCH may be entered as *CO ANY.

The various forms of the *COBI command are described in greater detail in the following text.

ANYBATCH Function

The operator uses the ANYBATCH function of the *COBI command when he knows that there is background processing time available and he has no more batch jobs to be run. In this case, he may want to begin processing of any jobs that COBI has prepared. He does this with the following command:

```
*COBI ANYBATCH
```

If the input data set currently attached to COBI has no jobs in it, the system responds with the message NO JOB AVAILABLE FOR BATCH. If the data set contains one or more jobs, the data sets are switched at the first opportunity: for example, after the current job has been completely entered into the data set. (The switching of the data sets occurs even if the conditions specified in the RDRQTY and RDRTIM parameters have not been met; see the description of the RESET function.)

Further action by the operator is not needed if the automatic mode of starting a COBI reader is in effect. The switching of the data sets causes the COBI reader to begin reading the appropriate input data set. The other input data set is then attached to COBI to receive more jobs.

If, however, the manual mode of starting the reader is in effect, the operator must start a reader after the data sets have been switched. The system issues a message which tells the operator what reader he is to start. This message appears on the OS/360 system operator's console; the operator must then issue, on the same console, the appropriate start command. (A reader may not be started from the command console.)

The following example shows the use of the ANYBATCH function with a system in the manual mode:

```
Command Console          OS/360 System Operator's Console

*COBI ANYBATCH           DIBEX021 10 JOBS ON SYSINA,START DIBRDRA
                         S DIBRDRA
READY
```

This sequence of operations causes DIBRDRA to begin reading the SYSINA data set. The SYSINB data set is attached to COBI to receive new jobs.

If the ANYBATCH function is used during system initialization and both input data sets are full, the DIBEX021 message is issued with blanks appearing in place of the number of jobs.

DSSTATUS Function

The operator uses the DSSTATUS function of the *COBI command to request the status of data sets associated with COBI jobs. Status may be requested for the data sets associated with all the jobs in the COBI index, for the data sets associated with a particular job or user number, or for the COBI input data sets, known as SYSINA and SYSINB. The valid forms of the command are:

```
*COBI DSSTATUS

*COBI DSSTATUS,userid[,userid]...[,userid]

*COBI DSSTATUS,job-number[,job-number]...[,job-number]

*COBI DSSTATUS,SYSINx
```

where

userid is a validated user number of the form aaannn

job-number is a COBI-assigned job number of the form #nnnnn and within the range from 2 to 32767

x is either A or B

If only DSSTATUS is specified, the status of the data sets associated with all the jobs in the COBI index is printed in job number sequence; if no jobs exist, a message is printed. If one or more user numbers are specified, the status of the data sets associated with all the jobs for the indicated users is printed in the order in which the user numbers appear on the command. If one or more job numbers are specified, the status of the data sets associated with only those jobs is printed in the order in which the job numbers appear on the command.

In all these cases, the output contains the number of the job and its associated data sets. Each data set is identified by type, which may be one of the following:

JCL Refers to the JCL for the indicated job

nPmm Refers to a SYSOUT data set which may be scanned and is defined in a cataloged procedure, where n is the number of the procedure and mm is the number of the data set

Unnn Refers to a user-defined data set which may be scanned, where nnn is the number of the data set

For cataloged procedure and user data sets, the volume serial number of the volume which contains the data set is also printed; for all data sets, the status is printed. The status may be one of the following:

ON-LINE — Indicates that the volume is mounted and the data set is available for scanning

OFF-LINE — Indicates either that the volume is not mounted, or, if the volume is mounted, that the data set is not available for scanning

JCL — For JCL data sets only, indicates that the JCL is available for scanning

A special form of the DSSTATUS function can be used to request the status of a COBI SYSIN data set by specifying either SYSINA or SYSINB. The output in this case consists of a list of all the jobs in the specified SYSIN data set. This list contains the job number and the user number associated with each job. The job numbers are listed in their order of submission to the SYSIN data set. If a job number is preceded by an asterisk, the job was cancelled.

The data set status information normally appears on the command console used to issue the request. However, the output may be sent to the high-speed printer by entering *COBI-P with the appropriate DSSTATUS request; in this case, *COBI-P must be entered as shown, with no abbreviation, but DSSTATUS may be abbreviated as DSS. In either case, the system responds with READY at the command console when the output has been completed. When the output is sent to the high-speed printer, the end of the output is indicated by END on the printer.

The following examples show the use of the *COBI command to request data set status. A sample of the appropriate output is also shown.

Example 1:

```
Operator:  *COBI DSSTATUS,#25,#3276,#1085

System:    JOBID:      SYSOUT:     VOL ID:     STATUS:

            #25        1P05        111111      ON-LINE
            #3276      U003        EXEC99      OFF-LINE
                       1P02        RTOSLK      OFF-LINE
            #1085      JCL
                       U001        222222      ON-LINE
                       2P01        111111      ON-LINE
            READY
```

Example 2:

```
Operator:  *CO DSS,IBM406,IBM789

System:    JOB ID:     SYSOUT:     VOL ID:     STATUS:
            IBM406
            #128       2P02        222222      ON-LINE
                       U001        222222      OFF-LINE
            #83        JCL
                       1P03        111111      ON-LINE
            IBM789
            #56
                       1P01        111111      ON-LINE
                       1P02        222222      ON-LINE
            READY
```

Example 3:

```
Operator:   *COBI DSS,SYSINA

System:     SYSINA       JOBID:       USER ID:

                          #56          IBM789
                         *#652         MMW586
                          #128         IBM406
                          #1085        KIM256
                         *#83          IBM496
                          #3265        KIM256
                          #102         IBM308
                          #25          IBM308
                         *#3276        MMW586
            READY
```

JOBSTATUS Function

The operator uses the JOBSTATUS function of the *COBI command to request the status of COBI jobs. Status may be requested for all jobs submitted for processing through COBI, for the jobs of a specific user, or for a specific job. The valid forms of the command are:

```
*COBI JOBSTATUS

*COBI JOBSTATUS,userid[,userid]...[,userid]

*COBI JOBSTATUS,job-number[,job-number]...[,job-number]
```

where

userid   is a validated user number of the form aaannn

job-number is a COBI-assigned job number of the form #nnnnn and within the range 2 to 32767

If only JOBSTATUS is specified, the status of all the COBI jobs in the COBI index is printed in job number sequence; if no jobs exist, a message is printed. If one or more user numbers are specified, the status of all jobs associated with the specified users is printed in the order in which the user numbers appear on the command. If one or more job numbers are specified, only the status of the indicated jobs is printed in the order in which the jobs numbers appear on the command; if a job with the number specified does not exist, an error message is printed.

In all cases, the output consists of the job number and the status of the job. The status may be one of the following:

CANCEL RQ   Indicates that the job is in OS/360 batch processing, but a request to cancel the job has been issued

COMPLETED   Indicates that execution of the job has been completed

JCL ERROR   Indicates that an error was detected when processing JCL statements

NOT DONE   Indicates that the job has been sent to OS/360 batch processing but it has not completed execution

One or more completion codes are printed to indicate the cause of each job termination. The codes may be either system or user completion codes. System codes are issued by OS/360 as described in the publication IBM System/360 Operating System: Messages and Codes; user codes, if any, are preceded by the letter U. Finally, the data sets associated with each job are listed by type, as follows:

JCL   Refers to the JCL for the indicated job

nPmm   Refers to a SYSOUT data set defined in a cataloged procedure, where n is the number of the procedure and mm is the number of the data set

Unnn   Refers to a user data set, where nnn is the number of the data set

The job status information normally appears on the command console used to issue the request. However, the output may be sent to the high speed printer by entering *COBI-P with the appropriate JOBSTATUS request; but in this case, *COBI-P must be entered as shown, with no abbreviation, but JOBSTATUS may be abbreviated as JOB. In either case, the system responds with READY at the command console after the output has been completed. When the output is sent to the high-speed printer, the end of the output is indicated by END on the printer.

The following examples show the use of the *COBI command to request job status. A sample of the appropriate output is also shown.

Example 1:

```
Operator:   *COBI JOBSTATUS,CAL428,IBM311,IB406

System:     JOB ID:      STATUS:      CODE:      DATA SET:
            CAL428
            #102       JCL ERROR              JCL,1P03,2P01
            #506       NOT DONE                   3P04,3P05
            IBM311
            #423       COMPLETED      U0333       4P02,4P03,U001
            #1056      NOT DONE               JCL,2P01
            #3000      CANCEL RQ
            PARAMETER,IB406    , INVALID ENTRY
            READY
```

Example 2:

```
Operator:   *CO JOB,#63,#145,#7200

System:     JOB ID:      STATUS:      CODE:      DATA SET:

            #63        COMPLETED   0C5,U0777      1P01,U005
            #145       COMPLETED        0C4    JCL,U007,2P01
            JOB ID #7200 HAS NO COBI INDEX RECORD
            READY
```

RESET Function

The operator uses the RESET function of the *COBI command to override initial startup parameters specified when CALL-OS was initialized. The mode of starting a reader may be changed as well as the conditions for switching readers. The valid forms of the command are:

```
*COBI RESET,AUTRDR=xxxxxx

*COBI RESET,RDRQTY=nnn
```

```
    *COBI RESET,RDRTIM=mmm
```

where

xxxxxx indicates the mode to be used to start a reader and must be either AUTO or MANUAL

nnn specifies the number of jobs to be accepted into a COBI input data set before switching takes place; the number must be within the range from 1 through 999

mmm specifies the number of minutes that are to elapse before switching takes place; the number must be within the range from 1 through 999

The AUTRDR option is used to put the COBI reader into either the automatic or manual mode. Automatic mode indicates that COBI is to start the appropriate reader after the input data sets are switched. Manual mode indicates that COBI is to notify the operator when a reader must be started and which one to start.

The RDRQTY option overrides the identical startup parameter which governs the frequency with which the input data sets are switched. Each time the specified number of jobs have been placed in the input data set attached to COBI, a switch is attempted in order to give the jobs to OS/360. At times, the alternate input data set may still be tied up with the COBI reader; in this case, COBI continues writing on the same input data set but makes the switch as soon as the reader has finished reading the alternate data set.

The RDRTIM option is another method of keeping jobs moving through the system. This option ensures that even if there are not many COBI jobs being submitted, no one job has to wait too long before being processed. After the specified number of minutes have elapsed, COBI attempts to switch input data sets and to initiate the reading of the data set which was attached to COBI.

Note: The RESET function of the *COBI should be used at the direction of the installation system programmer. More information on the COBI reader and the RDRQTY and RDRTIM initialization parameters is found in the publication CALL-OS Executive and Utilities Program Description Manual.

The following example shows the use of the RESET function of the *COBI command:

```
    Operator:  *COBI RESET,RDRQTY=5
    System:   READY
    Operator:  *COBI RESET,RDRTIM=10
    System:   READY
```

The input data set attached to COBI is switched either after five jobs have been placed in it or after ten minutes have elapsed since the data set was attached, whichever occurs first.

SCRATCH Function

The operator uses the SCRATCH function of the *COBI command to scratch one or all data sets associated with specific job and user number. The scratched space is made available to other programs. The data set is scratched only if the retention period has expried; a retention period of seven days is set when the data set is created and thereafter when the data set is scanned but not scratched. The valid forms of the command are:

```
    *COBI SCRATCH,userid,job-number

    *COBI SCRATCH,userid,job-number,JCL

    *COBI SCRATCH,userid,job-number,nPmm

    *COBI SCRATCH,userid,job-number,Unnn
```

where

userid — is a validated user number of the form aaannn

job-number — is a COBI-assigned job number of the form #nnnnn and within a range from 2 to a maximum determined by the size of the COBI index data set

JCL — specifies that only the JCL is to be scratched

nPmm — specifies that a SYSOUT data set defined in a cataloged procedure is to be scratched, where n is the number of the procedure and mm is the number of the data set

Unnn — specifies that a user-defined data set is to be scratched, where nnn is the number of the data set

If a data set is not specified, all SYSOUT data sets and the JCL, if it exists, associated with the specified user and job are scratched; the job is removed from the COBI index and the entry in the user's catalog is deleted. If either the JCL or a SYSOUT data set is specified, only that data set is scratched; if the scratched data set is the last data set associated with the job, then the job is removed from the COBI index and from the user's catalog. If the scratch request is successful, the system responds with READY.

The following examples show the use of the SCRATCH function of the *COBI command.

<u>Example 1</u>:

```
    Operator:  *COBI SCRATCH,IBM428,#58
    System:   READY
```

In this case, all data sets associated with job number #58 for user number IBM428 are to be scratched unless the retention period has not yet expired.

<u>Example 2</u>:

```
    Operator:  *COBI SCRATCH,KIM306,#476,JCL
    System:   READY
```

In this case, only the JCL associated with job number #476 for user number KIM306 is to be scratched (unless the retention period has not expired). Any other data sets associated with the job and user number are kept.

<u>Example 3</u>:

```
    Operator:  *COBI SCRATCH,DEV607,#8,2P4
    System:   READY
```

In this case, only the fourth data set in the second procedure of job number #8 submitted by user DEV607 is to be scratched (unless the retention period has not expired). Any other data sets associated with the job and user number are kept.

SUBMIT Function

The operator uses the SUBMIT function of the *COBI command to either inhibit or allow submission of jobs from a user terminal. The valid form of the command is:

```
*COBI SUBMIT=xxx
```

where

xxx   is either NO or YES

The NO option should be used to temporarily inhibit the submission of jobs. When this option is used, the operator should then issue the *WARN command to notify terminal users of the action and its duration. For example, it may be a temporary condition or it may be that the SUBMIT command cannot be used for the remainder of the day.

The YES option is used to permit the submission of jobs after the operator has previously used SUBMIT=NO to inhibit it. Job submission is automatically in effect when CALL-OS is initialized on a system built for COBI (provided NOCOBI was not specified in the startup deck).

If the abbreviated form of the command is used (*CO SUB), the default is YES. However, if SUBMIT= is specified, either YES or NO must be specified; if neither is present, an error message is issued.

The following example shows the use of the SUBMIT function of the *COBI command:

```
Operator:  *COBI SUBMIT=NO
System:   READY
Operator:  *WARN
System:   ENTER
Operator:  SUBMIT TEMPORARILY DISABLED - UP AGAIN AT 12:00
System:   READY
```

The submission of jobs has been temporarily inhibited. The operator notifies the terminal users with the *WARN command that they may submit jobs again at 12:00 at which time the operator must issue the following command:

```
*COBI SUBMIT=YES
```

*DATE COMMAND

The *DATE command allows the operator to enter today's date plus any other pertinent information. The operator may enter up to 24 characters plus carrier return. This information becomes an integral part of the heading line that is printed either at sign-on time or in response to a RUN, CATALOG, or LIST command. The format of the *DATE command is:

```
    *DATE
```

The system responds with ENTER and the operator enters the additional information.

The following example shows the use of the *DATE command to insert the date into the sign-on message:

```
    Operator:  *DATE
    System:   ENTER
    Operator:  WEDNESDAY, JUNE 30, 1971
    System:   READY
```

*DISABLE COMMAND

The *DISABLE command allows the operator to disable a single line or a set of contiguous lines. In either case, at least one parameter must be present. The lines for the command and communications consoles may not be disabled.

Disabling a Single Line

The operator may disable one line with the following format of the *DISABLE command:

```
*DISABLE nnn
```

where

nnn   is the logical line number of the line to be disabled

The following example shows the use of the *DISABLE command to disable the line whose logical line number is 35:

```
Operator:  *DISABLE 35
System:    READY
```

Disabling a Set of Lines

The operator may disable a set of contiguous lines. He does this by specifying a range of line numbers in the *DISABLE command, as follows:

```
*DISABLE nnn,mmm
```

where

nnn   is the logical line number of the first line in the set of lines to be disabled (that is, lowest number)

mmm   is the logical line number of the last line in the set of lines to be disabled

The following example shows the use of the *DISABLE command to disable a set of contiguous lines:

```
Operator:  *DISABLE 10,20
System:    READY
```

All logical lines between and including logical line numbers 10 and 20 are disabled.

*ENABLE COMMAND

The *ENABLE command causes the executive to enable terminal user lines. The operator may enable a single line, a set of contiguous lines, or all lines on the system with this command.

Enabling a Single Line

The operator may enable one line with the following format of the *ENABLE command:

```
    *ENABLE nnn
```

where

nnn   is the logical line number of the line to be enabled.

The following example shows the use of the *ENABLE command to enable the line whose logical line number is 35:

```
    Operator:  *ENABLE  35
    System:    READY
```

Enabling a Set of Lines

The operator may enable a set of contiguous lines. He does this by specifying a range of line numbers in the *ENABLE command, as follows:

```
    *ENABLE nnn,mmm
```

where

nnn   is the logical line number of the first line in the set of lines to be enabled (that is, the lowest number)

mmm   is the logical line number of the last line in the set of lines to be enabled

The following example shows the use of the *ENABLE command to enable a set of contiguous lines:

```
    Operator:  *ENABLE 10,20
    System:    READY
```

All lines between and including logical line numbers 10 and 20 are enabled.

Enabling All Lines

During the initialization of a CALL-OS system, the operator may use a special form of the *ENABLE command to enable all terminal user lines. (The lines for the command and communications console are enabled separately.) The format of the *ENABLE command is:

```
    *ENABLE
```

This form of the *ENABLE command does not require parameters or further action by the operator.

The following example shows the use of the *ENABLE command to enable all logical lines in the system:

```
Operator:  *ENABLE
System:    READY
```

<u>Note</u>: If a line has been taken out of service (that is, a message has been sent to the communications console telling the operator to busy out the data set), the *EN command is not effective for that line.

*IGNORE COMMAND

The *IGNORE command disables all lines which are not currently active. In addition, it prevents reenabling of a line when a current user hangs up. The effect of this command may be overridden by use of a *ENABLE command. The format of the *IGNORE command is:

```
*IGNORE
```

This command does not require any parameters or further action by the operator.

The following example shows the use of the *IGNORE command:

```
Operator:  *IGNORE
System:    READY
```

All inactive lines are disabled and no new users can sign on the system.

*MESSAGE COMMAND

The *MESSAGE command allows a one-line message to be inserted into the system. The one-line message is sent to users at sign-on time. The operator may enter a message of up to 74 characters plus carrier return. If the message is longer than 75 characters, the system responds with MESSAGE IS TOO LONG. The format of the *MESSAGE command is:

```
    *MESSAGE
```

The system responds with ENTER, and the operator enters the desired message. A previous message may be canceled by entering *MESSAGE and, in response to ENTER, depressing the RETURN key without entering any information.

The following example shows the use of the *MESSAGE command to transmit information to users when they sign on the system:

```
    Operator:  *MESSAGE
    System:   ENTER
    Operator:  CALL-OS UP UNTIL 1100, RUN ***NEWS FOR NEW HOURS
    System:   READY
```

*OFF COMMAND

The *OFF command disconnects all terminals and, after approximately one minute, terminates the CALL-OS job. The system prints a SYSTEM OFF message on the issuing command console and the communications console before shutting down. The format of the *OFF command is:

```
*OFF
```

This command does not require any parameters or further action by the operator.

The following example shows the use of the *OFF command to terminate system activity for the day:

```
Operator:  *OFF
System:    SYSTEM OFF AT 18:50
```

The system response also appears on the communications console. After a minute or two, the CALL-OS system is terminated and a message appears on the OS/360 system operator's console.

*REPORT COMMAND

The *REPORT command produces a listing of system statistics which can help the installation system programmer to adjust his system to the current processing load. This listing is sent to the data set defined by the SYSPRINT DD card in the startup deck. The format of the *REPORT command is:

```
    *REPORT
```

This command does not require parameters or further action from the operator.

Once the report has been sent, a message appears on the command console, as shown in the following example:

```
    Operator:  *REPORT
    System:    END OF STATISTICAL REPORT
```

The report itself is sent to the system output unit. The following is a sample of the type of information produced:

- Buffer requests
- Command counters
- Number of input and output lines
- Number of disk I/O's
- Number of total and available tracks for each user group
- Various other statistics that may be of interest in evaluating performance

See the publication CALL-OS Executive and Utilities Program Description Manual for a sample of the *REPORT output.

*STATUS COMMAND

The *STATUS command displays pertinent information about a terminal user. The operator may request this status either by logical line number or by user number. For either format, the output is:

- Terminal number (LINO)
- USER number (USNUM)
- Terminal type (TTYPE)
- Program name (PNAME) for a source program, blank for an object program
- Program size in characters (SOURC) for a source program, zero for an object program
- Number of lines (NLINE) for a source program, zero for an object program

  The SOURC and NLINE information is accurate only after a command which causes a sort to take place, for example, RUN, LIST, or SAVE.
- Language processor (LANG)
- A code indicating the user's status (STAT); for example, RUN, LIST, SAVE, PURGE, etc., as defined in Appendix A
- A code indicating the terminal status (TCHST), for example, reading, writing, idle, etc., as defined in Appendix A
- Bit flags representing conditions on terminal lines (TFLG1), as defined in Appendix A
- OS/360 IOS terminal communications switch (IOSW), as defined in Appendix A

Displaying Status by Logical Line Number

The operator may display the status of the user on a particular logical line with the following format of the *STATUS command:

```
*STATUS nnn
```

where

nnn   is the logical line number

The following example shows the use of the *STATUS command to display the status of the user on logical line number 5:

```
Operator:  *STATUS 5
System:    LINO     5
           USNUM    AAA001
           TTYPE    IBM 2741C
           PNAME    TESTPROG
           SOURC    234
           NLINE    17
           LANG     PL/I
           STAT     01
           TCHST    04
           TFLG1    00
           IOSW     00
```

Displaying Status by User Number

The operator may display the status of a particular user with the following format of the *STATUS command:

```
*STATUS aaannn
```

where

aaannn is the user number and consists of three letters aaa and three numbers nnn

The following example shows the use of the *STATUS command to display the status of the user whose number is IBM030:

```
Operator:   *STATUS IBM030
System:     LINO    10
            USNUM   IBM030
            TTYPE   IBM 2741E
            PNAME
            SOURC   0
            NLINE   0
            LANG    BASIC
            STAT    02
            TCHST   00
            TFLG1   00
            IOSW    00
```

*TELL COMMAND

The *TELL command allows the command console operator to transmit a message to another terminal. Only one message can be sent to one terminal at a time. The message is transmitted at the first logical break in the I/O sequence for the terminal. The operator may enter a message of up to 74 characters plus carrier return. If the message is longer than 75 characters, the message MESSAGE IS TOO LONG is printed at the command console. The command may be used to transmit messages to one of the following:

- A particular logical line
- A particular user
- The other command console
- The communication console

Special forms of the *TELL command may be used to either display a pending message from the global table on the command console or to erase a pending message in the global table.

Transmitting a Message to a Particular Line

The operator may transmit a message to the user on a particular logical line with the following format of the *TELL command:

```
*TELL nnn
```

where

nnn is the logical line number of the terminal which is to receive the message

The system responds with ENTER and the operator enters the message to be transmitted.

The following example shows the use of the *TELL command to transmit a message to a particular line:

```
Operator:  *TELL 25
System:   ENTER
Operator:  PLEASE CALL COMPUTER INSTALLATION (453-8146)
System:   READY
```

The message is sent to logical line 25 at the first break in the line's I/O sequence.

Transmitting a Message to a Particular User

The operator may transmit a message to a particular user with the following format of the *TELL command:

```
*TELL aaannn
```

where

aaannn is the user number of the user who is to receive the message; a user number consists of three letters aaa and three numbers nnn

The system responds with ENTER and the operator enters the message to be transmitted.

The following example shows the use of the *TELL command to transmit a message to a particular user:

```
Operator:  *TELL KIM408
System:   ENTER
Operator:  VOLID KIMTST NOT AVAILABLE, ANY INSTRUCTIONS ? JOB #28
System:   READY
```

The message is transmitted to the user whose number is KIM408 at the first break in the terminal's I/O sequence.

Transmitting a Message to the Command Console

The operator may transmit a message to the other command console with the following format of the *TELL command:

```
*TELL SYSLIB
```

The system responds with ENTER and the operator enters the message to be transmitted. If the sending command console operator is the only SYSLIB user, he receives his own message.

The following example shows the use of the *TELL command to transmit a message to the other command console:

```
Operator:  *TELL SYSLIB
System:   ENTER
Operator:  SYSTEM GOING DOWN AT 2100 HOURS
System:   READY
```

The message is transmitted to the other command console at the first break in the console's I/O sequence.

Transmitting a Message to the Communications Console

The operator may transmit a message to an online communications console with the following format of the *TELL command:

```
*TELL OP
```

The system responds with ENTER and the operator enters the message to be transmitted. If the communication console is inactive, the message is routed to the OS/360 system operator's console, preceded by the identifier DIBEX001.

The following example shows the use of the *TELL command to transmit a message to the communications console:

```
Operator:  *TELL OP
System:   ENTER
Operator:  ABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890-=!;',./±@#$%[&*()
System:   READY
```

The message is transmitted to the communications console at the first break in the console's output sequence.

Special Forms of the *TELL Command

The operator may display a pending message from the global table on the command console with the following format of the *TELL command:

```
*TELL D-MSG
```

If no message is present in the *TELL buffer in the global table, the system responds with NO MESSAGE PENDING; otherwise, the message itself is printed.

The operator may erase a pending message in the global table with the following format of the *TELL command:

```
*TELL E-MSG
```

If no message is present, the system responds with NO MESSAGE PENDING; otherwise, the system responds with READY after erasing the message.

*USERS COMMAND

The *USERS command prints the total number of users that are presently online either for the entire system or for a range of line numbers.

Total of All Users on the System

The operator may print the total number of users on the system with the following format of the *USERS command:

```
*USERS
```

The following example shows the use of the *USERS command to obtain the total number of users on the system:

```
Operator:  *USERS
System:    83 USERS AT 12:35
```

The total of 83 does not include the command and communications consoles.

Total of Users for a Set of Lines

The operator may print the total numbers of users for a range of line numbers with the following format of the *USERS command:

```
*USERS nnn,mmm
```

where

nnn   is the logical line number of the first line in the set

mmm   is the logical line number of the last line in the set

If the range specified covers the logical lines of the command and/or communications consoles, the count includes the command console(s) but not the communications console.

The following example shows the use of the *USERS command to obtain the total number of users on logical lines 10 through 20:

```
Operator:  *USERS 10,20
System:    11 USERS AT 12:45
```

*VALIDATE COMMAND

The *VALIDATE command validates a new user number onto the system and assigns a password to the user. If the user number has already been validated, this command can be used to change the password. A special form of the *VALIDATE command allows users in subscription group to share files within the group.

Validating a New User Number

The operator may validate a new user number with the following format of the *VALIDATE command:

```
*VALIDATE aaannn,password
```

where

aaannn — is the user number to be validated and consists of three letters aaa and three numbers nnn

password — is up to eight characters and may consist of any combination of letters, numbers, and special characters

If the specified user number has not already been validated, the number is validated and the specified password becomes that user's password. If, however, the user number has already been validated, a message is printed; the operator must then reissue the *VALIDATE command, using the format for changing the password.

The following example shows the use of the *VALIDATE command to validate a new user number:

```
Operator:  *VALIDATE MMW483,D816U384
System:   READY
```

User number MMW483 is validated and D816U384 is assigned as that user's password.

Changing a Validated User's Password

The operator may change the password for a user whose number is already validated with the following format of the *VALIDATE command:

```
*VALIDATE aaannn,password,C
```

where

aaannn — is a user number that has already been validated and consists of three letters aaa and three numbers nnn

password — is the new password to be assigned to the user number and is up to eight characters; the password may consist of any combination of letters, numbers, and special characters

C — indicates that the specified password is to replace the current password assigned to user number aaannn

If the user specified number has already been validated, the password originally assigned to that user is changed. If, however, the user number has not already been validated, a message is printed and the operator must then reissue the *VALIDATE command, omitting the C; in this case, the new user number is validated and the specified password is assigned to the user.

The following example shows the use of the *VALIDATE command to change a password:

```
Operator:  *VALIDATE MMW483,D816U484,C
System:    READY
```

Since user number MMW483 is already validated, the password assigned to that number is changed to D816U484.

Validating a *Directory

A user may make his files available to other members of his group by pooling the files in a *Library. In order for this library to be available to a given set of users, the '00' user must be validated for that group. The *Library is then available to only those users whose user numbers have the same first four characters. This special form of the *VALIDATE command is:

```
*VALIDATE aaan00
```

where

aaan00 is a special user number and consists of three letters aaa, one variable number n, and the numerals 00

When this form of the *VALIDATE command is used, a maximum of 99 users (one subscription group) may pool information in the appropriate *Library. A password should not be supplied; if one is supplied, it is ignored.

The following example shows the use of the *VALIDATE command to allow the members of a particular subscription group to share a library:

```
Operator:  *VALIDATE AAA300
System:    READY
```

A *Library is now available to any user whose number is within the range AAA301 through AAA399.

*WARN COMMAND

The *WARN command allows a warning message to be inserted in the system. This message is transmitted to all online users at the first logical break in each terminal's I/O sequence. The operator may enter a message of up to 74 characters plus carrier return. If the message is longer than 75 characters, the system responds with MESSAGE IS TOO LONG. The format of the *WARN command is:

```
*WARN
```

The system responds with ENTER and the operator enters the desired message. A previous warning message may be canceled by entering *WARN and, when the system responds with ENTER, depressing the RETURN key without entering a message.

The following example shows the use of the *WARN command to notify all users that the system is to be shut down temporarily:

```
Operator:  *WARN
System:    ENTER
Operator:  SYSTEM GOING DOWN AT 11:00 FOR 30 MINUTES
System:    READY
```

The warning message is transmitted to each terminal at the first break in the terminal's I/O sequence.

OS/360 SYSTEM OPERATOR'S CONSOLE MESSAGES

During online operations, the OS/360 system operator's console receives messages from the initialization routines and from the executive in the form of user ABEND messages. In addition, when COBI is used, the operator may receive messages pertaining to actions he must perform (for example, start a reader to read a COBI input data sets). Finally, if the communications console for CALL-OS is inoperative, messages intended for it appear on the OS/360 system console.

During offline operation, this console receives utility ABEND codes and messages, as well as COBI messages if the COBI reader and/or writer programs are executed.

INITIALIZATION MESSAGES

The CALL-OS initialization routines perform validity checks on the data base and other system components. Many potential problems are diagnosed before users sign on the system. The action taken upon discovering an error depends on the severity of the error. If only one user group is affected, system initialization will continue with an error message indicating the extent to which the user group's operation will be affected. If it is likely that the initialized configuration will not match the one desired by the user, the initialization routine will issue an explanatory message and ABEND.

The primary use of these messages will be evident during the early stages of system installation. It is anticipated that once an installed CALL-OS system has been brought up successfully, initialization error messages should rarely occur. In some cases, the indicated problem can be corrected by modifying the specified JCL card or parameter. (Appendix C summarizes the JCL and parameters in the startup deck.) In other cases, however, only a trained systems programmer or analyst can properly diagnose and correct the situation.

All messages printed on the OS/360 operator's console are preceded by a message identification of DIBINxyy

where

x identifies the initialization module which issued the message and may be one of the following values:

x	Module	x	Module
0	N#LINIT	4	N#OPENUG
1	N#INIT	5	N#RESM
2	N#CHKDDS	6	N#POOL
3	N#OPENLG	7	N#BLDTBL

yy is a sequence number within the messages issued by that module.

If the error is severe enough to cause an ABEND, the user code associated with that ABEND will be 1xyy where xyy has the same meaning as above and is associated directly with the error messages with one exception. For error messages DIBIN202, DIBIN204, DIBIN207, DIBIN212, and DIBIN213, a composite ABEND code of 1299 is used since more than one error condition can occur prior to abending the job.

OS/360 System Console Initialization
Messages and Explanations

DIBIN001 CALL-OS INITIALIZED. NJA=nnnnnn,OJA=mmmmmm.

This is the normal initialization message which will be printed when the system is ready to begin communication with the command and communications consoles. The size of the new job area is specified by nnnnnn and the size of the old job area is specified by mmmmmm. Each CALL-OS installation will have different acceptable values for these two parameters, and the operator should carefully check to ensure that the values printed fall within these installation limits. A value of 000000 for the old job area is possible and indicates that there is no old job area.

DIBIN002 INSUFFICIENT MAIN STORAGE TO BEGIN INITIALIZATION.

The core storage area allocated to CALL-OS is far too small. A probable cause is that the job was initialized in the wrong MFT partition or that the region parameter was not properly specified on the JOB or EXEC card for MVT. Restart the job with adequate storage.

DIBIN003 INSUFFICIENT H-1 STORAGE TO BEGIN INITIALIZATION.

The LCSRES parameter on the EXEC card specifies that the new and old job areas are to be located in Hierarchy 1 (H-1) storage. The amount of H-1 storage available to the initialization routines was found to be less than the 53,248 bytes required for a minimum new job area.

Ensure that the task area allocated matches the H-1 options specified, and that the CLASS specified on the JOB card is correct. Then restart the job.

DIBIN004 INSUFFICIENT MAIN STORAGE FOR A MINIMUM NEW JOB AREA.

At the end of system initialization, there was too little storage available to allocate the minimum new job area of 53,248 bytes. Add enough core storage to the allocated area to bring the new job area up to the size desired.

DIBIN005 INSUFFICIENT H-1 STORAGE FOR MINIMUM NEW JOB AREA.

The LCSRES parameter on the EXEC card specifies that the new and old job areas are to be located in Hierarchy 1 storage. After all other areas have been allocated, it was found that less than the 53,248 bytes of H-1 storage was available to allocate a minimum new job area.

Increase the amount of storage available in H-1 and restart the job.

DIBIN006 nnnnnn EXCESS BYTES OF MAIN/H1 STORAGE ALLOCATED TO CALL-OS.

The storage area allocated to CALL-OS in the indicated storage hierarchy exceeded the maximum usable by the system by nnnnnn bytes. The initialization process will proceed, but on the next running of the system one of the following actions should be taken:

- Decrease the amount of storage allocated in the hierarchy indicated.

- Specify a larger number of input and output buffers.

- Specify more resident routines via the RESMODS DD card.

DIBIN007 SYSIN CONTAINS TOO MUCH PARAMETER INFORMATION.

More than 400 characters of parameter information have been supplied either in the SYSIN data set alone, or between the parameter field on the startup deck EXEC statement and the SYSIN data set. Correct the SYSIN data set and restart the job.

DIBIN008 INSUFFICIENT MAIN STORAGE FOR ABEND/ABDUMP.

The job areas have been allocated in H-1 storage, but the 6144 bytes necessary to ensure successful ABEND dumps are not available in main storage. Initialization will continue, but it is probable that a complete dump will not be produced should an ABEND occur.

DIBIN009 CALL-OS SYSTEM BUILT NOT COMPATIBLE WITH OS/360 SYSTEM

An attempt has been made to initialize a CALL-OS system on an OS/360 configuration (either MFT or MVT) for which CALL-OS was not built.

DIBIN101 OS/360 ASSIGNED A PROTECT KEY OF ZERO TO CALL-OS.

If this message appears, a serious interface problem has occurred. The ABEND dump produced should be immediately given to the IBM Program Systems Representative, along with the accompanying console sheet representing the previous OS/360 activity which has occurred during the current IPL of OS/360. In order to start the CALL-OS job, it will be necessary to re-IPL OS/360.

DIBIN102 INVALID KEYWORD OR VALUE FOR cccccccccccccccc.

An invalid keyword or parameter has been detected in the parameter information supplied on the EXEC card or in the SYSIN data set. The text which contains the error is contained in the cccccccccccccccc field of the message.

DIBIN103 UNABLE TO OPEN CALL-OS INDEX.

The INDEX DD card is missing from the startup deck or some problem has made it impossible for the open routine to successfully process it. Correct the startup deck and restart the job.

DIBIN104 I/O ERROR READING CALL-OS INDEX.

A serious error has occurred while attempting to read the INDEX. The systems programmer or analyst should be contacted immediately since the system cannot be successfully initialized without the presence of a valid INDEX data set. This message will also appear if the INDEX contains only a file mark.

DIBIN105 INVALID FIRST ENTRY IN CALL-OS INDEX.

The INDEX has been successfully read but the first entry indicates that there are no compilers in the system. The system cannot be initialized until this situation is corrected.

DIBIN106 nnnnnnnn IN COMTSL PARAMETER INVALID LANGUAGE NAME.

The language name specified by nnnnnnnn does not have a matching entry in the CALL-OS language table. Correct the name in the parameter and restart the job.

DIBIN107 TTT OF SHRTSL IS NOT LESS THAN THE OLD JOB TIME SLICE.

The time allocated to background work is derived from the old job time slice. Correct the value specified in the SHRTSL parameter and restart the job.

DIBIN108 CALL-OS MARKED FOR TIME-SLICING BY OS/360.

The TCB for CALL-OS is marked for OS/360 time slicing. CALL-OS may not be time sliced. To restart CALL-OS, change the PRTY (MVT) or CLASS (MFT) and reenter the startup deck into the job stream.

DIBIN109 INVALID CHARACTER STRING AT END OF PARM FIELD.

The PARM field does not end with a valid keyword or value. Correct the PARM field and restart the job.

DIBIN110 CALL-OS MARKED ROLLOUTABLE BY OS/360

The TCB for CALL-OS is marked rolloutable; however, CALL-OS may not be rolled out by OS/360. Restart the job specifying ROLL=(NO,NO) on either the JOB or EXEC card of the startup deck.

DIBIN111 UNABLE TO OBTAIN CORE FOR COBI GLOBAL TABLE

Check to ensure that the CALL-OS job has been initiated in the desired task area. For MVT, ensure that the desired task area size was specified correctly. Correct the task area specification and restart the job.

If Hierarchy 1 storage is being used, check the LCSRES parameter specification for a G to determine in which hierarchy the shortage has occurred.

DIBIN112 SUBTASKING OPTION NOT AVAILABLE

The user is attempting to initialize CALL-OS on an MFT system which does not allow subtasking. Subtasking for MFT is specified during system generation with the OPTIONS=ATTACH parameter of the SUPRVSOR macro instruction.

DIBIN113 OSCLASS EQUAL TO CBCLASS

The OSCLASS parameter specifies the same output class as the CBCLASS parameter. This is not permissible because the CBCLASS output class is reset to the OSCLASS output class for processing by an output writer. Correct the OSCLASS parameter and restart the job.

DIBIN201 SEQUENCE ERROR IN INDEX AT ENTRY dddddddd.

CALL-OS will not be brought up until the INDEX has been corrected. Initialization of the system will be discontinued.

DIBIN202 MISSING DD CARD FOR GROUP dddddd.

If a user group is to be brought up, all DD cards for that group must be included in the startup deck. System initialization will continue to check the remaining entries, but the system will not be initialized.

DIBIN203 xxxxxxxx HAS NO MATCHING ENTRY IN INDEX AND IS NOT IN DD EXCEPTION TABLE.

The DD card specified in xxxxxxxx is unknown to the CALL-OS system and will be ignored. Correct the DD card to reflect the desired name or remove the DD card from the startup deck for future runs.

DIBIN204 OVERLAPPING ALTERNATE AND PRIMARY CLUSTER USER GROUPS BEING OPENED FOR GROUP xxxxxx.

DD cards have been supplied for a user range xxxxxx in both primary and alternate clusters. Remove the unwanted DD card from the JCL deck and restart the job.

DIBIN205 NO SYSTEM OR USER GROUP DATA SETS BEING OPENED.

Initialization cannot proceed without at least one system or user group being opened. Add the necessary cards to the startup deck and restart the job.

DIBIN206 NO SYSTEM GROUP BEING OPENED.

No SYSGRPnn DD card was included in the startup deck and the system will not be initialized. Include a system group DD card in the startup deck and restart the job.

DIBIN207 TRYING TO OPEN TWO SYSTEM GROUPS.

DD cards have been included in the startup deck for both the primary and alternate system group cluster. Only one cluster is allowed. Remove the cards for either the alternate or primary cluster and restart the job.

DIBIN208 NO TERMINAL DD CARDS PROVIDED.

No DD cards were included in the startup deck to allocate terminal devices and initialization cannot proceed. Include the appropriate cards and restart the job.

DIBIN209 UNIDENTIFIABLE INDEX ENTRY FLAGGED FOR OPEN. ENTRY NO.=nnn.

The appearance of this message probably indicates that something is seriously wrong with the INDEX. The ABEND dump produced should be shown immediately to the systems programmer or analyst responsible for the system.

DIBIN210 NO ENTRY IN LANGUAGE TABLE FOR COMPILER cccccccc.

This message may indicate a misspelled compiler DD name. If the DD name is correct, ensure that it has been entered properly in the INDEX. Initialization will continue, but the indicated DD card will be ignored.

DIBIN211 NO USER GROUP OR SYSTEM DATA SETS BEING OPENED.

Initialization cannot proceed without at least one data set being opened. Add the necessary cards to the startup deck; restart job.

DIBIN212 MISSING DD CARD FOR TWO PHASE COMPILER xxxxxxxx.

The named compiler is a multiphase processor. A DD card must be supplied for each phase if the compiler is to be used. Supply the missing DD card and restart the job.

DIBIN213 DSNAME DOES NOT MATCH INDEX - dddddddd.

The dsname specified on the DD statement dddddddd does not match the dsname as specified in the INDEX. Specify the correct dsname and restart the job.

DIBIN214 DUPLICATE DDNAME - dddddddd.

The ddname dddddddd appears more than once in the startup deck. Remove the duplicate DD statement and restart the job.

DIBIN215 ERROR ENCOUNTERED READING JOB FILE CONTROL BLOCK FOR
DDNAME=dddddddd.

The job file control block for the indicated DD statement (dddddddd) could not be read. The appearance of this message indicates a serious system problem. The ABEND dump produced should be brought to the attention of the systems programmer immediately.

DIBIN216 NO COMPILERS BEING OPENED

No compiler DD cards were provided. Add necessary cards to startup deck and restart job.

DIBIN217 COBI OPTION DD STATEMENTS NOT SUPPLIED

If COBI is to be active for this session of CALL-OS, the following DD statements must be present in the startup deck: CBNDX, CBJCL, CBSYSINA, CBSYSINB, and SYSJOBQ. Correct the startup deck and rerun the job.

DIBIN218 NO SCAN DD STATEMENTS SUPPLIED, SCAN AND SCRATCH INACTIVE

The startup deck did not contain any SCANxx (where xx is a unique two-character identifier) DD statements to define those volumes which are to contain data sets to be scanned at the user's terminal. The SCAN and SCRATCH commands cannot be used for this session of CALL-OS.

DIBIN219 ATTEMPT TO OBTAIN DEVICE ATTRIBUTES FAILED - DDNAME=dddddddd

The DEVTYPE macro instruction was used to request the device attributes for the data set defined by the specified DD statement (dddddddd). The request failed. This indicates a serious problem and the ABEND dump produced should be brought to the attention of the installation system programmer immediately.

DIBIN220 DEVICE TYPE NOT SUPPORTED FOR SCANNING - DDNAME=dddddddd

The specified DD statement (dddddddd) describes a device which is not supported for scanning of COBI job output. The DD statement is ignored.

DIBIN221 DATA SET NOT AVAILABLE - DDNAME=ddddddd

CALL-OS attempted to enqueue the data set defined by the indicated DD statement; however, the data base utility is enqueued on the data set. CALL-OS cannot be initialized until the utility either completes execution or is canceled.

DIBIN301 INSUFFICIENT CORE STORAGE ALLOCATED FOR DCB'S.

Check to ensure that the CALL-OS job has been initiated in the desired task area. For MVT, ensure that the desired task area size was specified correctly. Increase the task area specifications and restart the job.

If Hierarchy 1 storage is being used, check the LCSRES parameter specification for a D to determine in which hierarchy the shortage has occurred.

DIBIN302 TWX DD COULD NOT BE OPENED.

The TWX DD card provided could not be successfully processed by open. Check to ensure that the card or cards provided are properly concatenated and that the UNIT parameters are correct; restart job.

DIBIN303 T2741 DD COULD NOT BE OPENED.

The T2741 DD card provided could not be successfully processed by open. Check to ensure that the card or cards provided are properly concatenated and that the UNIT parameters are correct; restart job.

DIBIN304 TTY OR 2741 UCB NOT COMMUNICATIONS TYPE.

The DD cards provided for terminals contained a device address which was not specified as a communications device during system generation. Correct the cards and restart the job.

DIBIN305 T2741E DD COULD NOT BE OPENED.

The T2741E DD card provided could not be successfully processed by the open routine. Check to ensure that the card or cards provided are properly concatenated and that the UNIT parameters are correct; restart job.

DIBIN401 INSUFFICIENT CORE STORAGE ALLOCATION FOR DCB'S.

Check to ensure that the CALL-OS job has been initiated in the desired task area. For MVT, ensure that the desired task area size was specified correctly. Increase the task area specifications and restart the job.

If Hierarchy 1 storage is being used, check the LCSRES parameter specification for a D to determine in which hierarchy the shortage has occurred.

DIBIN404 UNABLE TO OPEN ddddddd DATA SET.

The data set with a DD name of ddddddd could not be successfully processed by the open routine. If the data set is part of a user group, the entire user group will not be brought up for this run. For the SYSPRINT data set, initialization continues but the *REPORT and *COBI-P functions will be inoperative. For all other data sets, an ABEND will result.

DIBIN407 ALLOC RECORD IDENT INCORRECT. DDNAME=dddddddd.

The first record for the named data set has an improper format. Initialization will continue but will not include this user group. The data set indicated must be reformatted before it can be used.

DIBIN408 USER NO. IN ALLOC RECORD NOT EQUAL TO INDEX DDNAME=dddddddd.

The user number range in the allocation record for the named data set does not match the user number range in the index. Initialization will continue but the condition should be corrected as soon as possible.

DIBIN410 USER GROUP DATA SET DEPLETED. DDNAME=dddddddd.

All available space in the named data set has been allocated. Initialization will continue. If all of the data sets in a user group are depleted, the installation will probably want to obtain additional disk storage space for the user group. This can be done by adding additional data sets to the group, and/or by running the REORGANIZE function of the data base utility to release purged disk space from user files.

DIBIN411 ERROR RETURNED FROM OS/360 CONVERT ROUTINE DDNAME=dddddddd.

Unable to successfully convert from TTR to MBBCCHHR format for the named data set. For user group data sets, initialization continues with the data set treated as fully allocated; reformat the data set to allow for its future use. For COBI data sets, initialization terminates; to initialize CALL-OS immediately, specify the NOCOBI parameter and restart the job.

DIBIN412 I/O ERROR READING ttttttttttt DDNAME=dddddddd.

An I/O error occurred while reading the named data set in the type of allocated record indicated by ttttttttttt. The data set extents are set to a fully allocated condition and initialization continues. Reformat the data set to allow its future use.

DIBIN413 K OR D LENGTH ERROR ON ttttttttttt DDNAME=dddddddd.

The format of the named CALL-OS data set is incorrect. The data set extents are set to indicate a fully allocated condition and initialization continues. Reformat the data set to allow for its future use.

DIBIN414 SPECIFIED SYSPRINT BLKSIZE NOT MULTIPLE OF 133.

The block size specified in the DCB parameter of the SYSPRINT DD card was not a multiple of 133. The default block size of 665 has been used.

DIBIN415 SPECIFIED SYSPRINT BLKSIZE GREATER THAN 1995.

The block size specified in the DCB parameter of the SYSPRINT DD card was greater than maximum value of 1995. A block size of 1995 has been used.

DIBIN416 USER GROUP INDEX ENTRIES NOT CONTINUOUS. USER GROUP=uuuuuu.

The entries in the INDEX for user group uuuuuu are not in relative data set number order. The INDEX must be corrected before the system can be initialized. The user group in question may be temporarily removed from the startup deck and restart attempted.

DIBIN417 INSUFFICIENT CORE STORAGE ALLOCATED FOR COBI BIT TABLES

Check to ensure that the CALL-OS job has been initiated in the desired task area. For MVT, ensure that the desired task area size was specified correctly. Increase the task area specification and start the job.

If Hierarchy 1 storage is being used, check the LCSRES parameter specification for a G to determine in which hierarchy the shortage has occurred.

DIBIN418 INVALID COBI INDEX

The data set defined by the CBNDX DD statement is not a valid formatted COBI index data set. Verify that the DD statement is correct. If it is, the COBI index data set must be reformatted and the installation system programmer should be notified. To initialize CALL-OS immediately, COBI may be made inactive by specifying NOCOBI in the run-time options for initialization and restarting the job.

DIBIN419 I/O ERROR READING COBI DATA SET - DDNAME=dddddddd

A permanent I/O error was encountered while reading the data set defined by the specified DD statement (dddddddd). The data set may have to be reformatted and the installation system programmer should be notified. To initialize CALL-OS immediately, COBI may be made inactive by specifying NOCOBI in the run-time options for initialization and restarting the job.

DIBIN420 ATTEMPT TO OBTAIN DEVICE ATTRIBUTES FAILED - DDNAME=dddddddd

The DEVTYPE macro instruction was used to request the device attributes for the data set defined by the specified DD statement (dddddddd). The request failed. This indicates a serious problem and the ABEND dump produced should be brought to the attention of the installation system programmer immediately.

DIBIN421 ERROR ENCOUNTERED READING JOB FILE CONTROL BLOCK FOR
DDNAME=dddddddd

The job file control block for the indicated DD statement (dddddddd) could not be read. The appearance of this message indicates a serious system problem. The ABEND dump produced should be brought to the attention of the installation system programmer immediately.

DIBIN422 INVALID COBI JCL DATA SET

The data set defined by the CBJCL DD statement is not a valid formatted COBI JCL data set. Verify that the DD statement is correct. If it is, the COBI JCL data set must be reformatted and the installation system programmer should be notified. To initialize CALL-OS immediately, COBI may be made inactive by specifying NOCOBI in the run-time options for initialization.

DIBIN423 SYSINA AND SYSINB DATA SETS UNAVAILABLE FOR USE - REPLY 'CONT'
OR 'CANCEL'

The SYSINA and SYSINB data sets either have not been initialized or have not been processed by the COBI reader (DIBRDRA or DIBRDRB, respectively). If the data sets have not been processed by the reader, the following options are available to the operator:

1. A reply of 'CONT' indicates that the data set last used by COBI is to be reattached for use and that the other data set is to be processed by the appropriate COBI reader. Another message (see DIBIN424) indicates which reader the operator must start.

2. A reply of 'CANCEL' indicates that CALL-OS is to issue ABEND 1423 and terminate processing. The operator may then start one or both COBI readers to process the appropriate data set(s).

If the data sets have not been initialized, the installation system programmer should be notified to format the data sets. In this case, the operator should reply 'CANCEL'.

DIBIN424 SYSINx NOW ATTACHED TO COBI. START DIBRDRy

This message is issued either after the operator replies 'CONT' to message DIBIN423 or if, from a previous CALL-OS session, a COBI SYSIN data set was created but not processed by the COBI reader. The message informs the operator which data set has been attached to COBI to receive submitted jobs and which reader he is to start to process the other data set. The operator must now start the appropriate reader. If x is A, then y is B; if x is B, then y is A.

DIBIN425 DIBWTR ACTIVE. STOP DIBWTR AND REPLY 'CONT', OR REPLY 'CANCEL'.

A CALL-OS system with COBI is being initialized and the COBI writer program DIBWTR is still processing COBI jobs. Since execution of DIBWTR is required only when COBI is not active, the operator must decide whether to continue or terminate the CALL-OS initialization process. To continue, he stops DIBWTR by issuing the appropriate STOP command and then replies 'CONT'. Otherwise, he replies 'CANCEL' and the initialization process is terminated.

DIBIN426 DIBRDR ACTIVE - SYSIN DATA SET UNAVAILABLE. REPLY 'WAIT' OR 'CANCEL'.

A CALL-OS system with COBI is being initialized and neither COBI input data set is available to receive jobs. The COBI reader program is active and reading one of the data sets. The operator may request that the initialization process wait until the reader has released the data set, in which case he replies 'WAIT'. Otherwise, he replies 'CANCEL' and the initialization process is terminated; the reader continues processing COBI jobs.

DIBIN427 VOLID TABLE EXCEEDS MAXIMUM

The maximum number of entries permitted in the COBI volume identification table is 254. Before the CALL-OS system can be initialized, U#5RINIT must be executed to reinitialize the COBI data sets.

DIBIN428 INITIAL DEVICE TYPE MODIFIED - VOLID=xxxxxx

The original device type of the specified volume is not the same as the device type on which the volume is now residing. To initialize the CALL-OS system immediately, remove the SCANxx DD statement for this volume from the startup deck and restart the job.

DIBIN429 I/O ERROR WRITING COBI VOLID TABLE

A permanent I/O error occurred while attempting to update the COBI volume identification table. The ABEND dump produced should be brought to the attention of the installation system programmer immediately. To initialize the CALL-OS system, specify the NOCOBI parameter in the startup deck and restart the job.

DIBIN430 MAXDCB VALUE EXCEEDS NUMBER OF LINES

The value specified for the MAXDCB parameter exceeds the number of lines defined for this session. Initialization continues. The value used is the number of lines.

DIBIN431 INSUFFICIENT MAIN STORAGE ALLOCATED FOR DEB'S.

MFT uses main storage in the CALL-OS partition for data extent blocks. The actual amount used depends on the number of users to be permitted to scan data sets concurrently (MAXDCB parameter). Increase the amount of storage in the CALL-OS partition accordingly and restart the job. To initialize CALL-OS immediately without COBI, specify the NOCOBI parameter and restart the job.

DIBIN432 INVALID SYSJOBQ DATA SET

The data set defined by the SYSJOBQ DD statement is not a valid OS/360 system job queue data set. Correct the DD statement and restart the job. To initialize the system immediately without COBI, specify the NOCOBI parameter and restart the job.

DIBIN433 INVALID CBSYSIN DATA SET. DDNAME-dddddddd.

The data set described by the specified DD statement is not a valid formatted COBI input data set. Ensure that the DD statement is correct; if the DD statement is correct, the data set must be reformatted. The ABEND dump produced should be brought to the attention of the installation system programmer immediately. To initialize CALL-OS without COBI, specify the NOCOBI parameter and restart the job.

DIBIN434 IDENTICAL COBI SYSIN DATA SETS

The same data set is described by both the CBSYSINA and CBSYSINB DD statement. Correct the DD statements and restart the job. To initialize CALL-OS without COBI, specify the NOCOBI parameter and restart the job.

DIBIN501 SYNTAX ERROR IN RESMODS LIST.

The RESMODS list has an invalid format. Check the RESMODS DD card to ensure that it is pointing to the desired member of SYS1.PROCLIB. Each card image in the member pointed to must contain a list of module names, starting in column 1 of each card image, separated by commas, with no intervening blanks, and not extending beyond column 71. No continuation indicator is required in column 72.

To bring the system up immediately, remove the RESMODS DD card. If a completely resident system is desired, also remove the OVLY DD card. If a completely nonresident system is desired, leave the OVLY DD card in the startup deck.

DIBIN502 ERROR IN RESMODS HIERARCHY PARAMETER.

A parenthesized expression other than (0) or (1) followed a module name in the RESMODS list. Correct the list and restart the job. To bring the system up immediately, remove the RESMODS DD card. If a completely resident system is desired, also remove the OVLY DD card. If a completely nonresident system is desired, leave the OVLY DD card in the startup deck.

DIBIN503 MODULE IN RESMODS LIST NAMED mmmmmmmm IGNORED. UNKNOWN TO SYSTEM.

The module name mmmmmmmm was included in the RESMODS list but cannot be found in the system. Initialization will continue, but a check should be made to ensure that this message does not indicate a misspelled name in the RESMODS list.

DIBIN504 MODULE NAMED mmmmmmmm IGNORED, NOT IN SPECIFIED JOBLIB.

Initialization will continue, but this message may indicate a serious system problem. If the module named is necessary for successful system operation (that is, not a temporary testing or development module) the situation should be corrected and the system restarted.

DIBIN505 INSUFFICIENT CORE STORAGE ALLOCATED FOR RESIDENT MODULES.

Check to ensure that the CALL-OS job has been initiated in the desired task area. For MVT, ensure that the desired task area size was specified correctly. Increase the task area specifications and restart the job.

If Hierarchy 1 storage is being used, check the heirarchy specifications in the RESMODS list to determine in which hierarchy the shortage occurred.

Note: The terminal translate tables are loaded the same as potentially resident modules. However, the default hierarchy for these tables is the hierarchy specified for pots (24-byte buffers).

DIBIN506 NO SPACE TO WRITE MODULE mmmmmm IN OVLY DATA SET. REST OF MODULES MADE RESIDENT.

Not enough space was available in the OVLY data set. Since one cylinder is sufficient for this data set, the occurrence of this message may mean serious system problems.

Initialization will continue with the remaining modules being made resident, but to obtain the exact resident configuration desired, the OVLY data set should be reallocated.

DIBIN507 I/O ERROR WRITING MODULE mmmmmmmm IN OVLY DATA SET. REST OF MODULES MADE RESIDENT.

An irrecoverable I/O error occurred while writing nonresident modules to the CALL-OS OVLY data set. Initialization will continue with the remaining modules being made resident, but to obtain the exact resident configuration desired, the OVLY data set should be reallocated.

DIBIN508 MODULE mmmmmmmm MADE RESIDENT. TOO LONG FOR OVERLAY.

The named module exceeds the one track maximum size for nonresident modules. The CALL-OS system, as released, contains no such modules. If no user modifications have been made to the system, this message indicates a serious system problem.

DIBIN509 RESMODS LIST SPECIFIED 'NONRES' BUT NO OVLY DD CARD PROVIDED.

NONRES in the RESMODS list indicates that all potentially resident modules are to be nonresident. If this configuration is desired, supply the OVLY DD card and restart the job.

DIBIN510 "IDENTIFY" FAILED FOR MODULE xxxxxxxx

An IDENTIFY macro instruction was issued for the specified module (xxxxxxxx). The load module could not be found in main storage. The appearance of this message indicates a serious system problem. The ABEND dump produced should be brought to the attention of the installation system programmer immediately.

DIBIN601 I/O ERROR READING EQUIVALENCY RECORD FOR USER NO. uuuuuu.

The **Directory for this user will not be updated. The operator may wish to supply the DD cards to bring up the alternate cluster (if any) for this user.

DIBIN602 INVALID USER NO. RANGE IN EQUIVALENCY RECORD FOR USER NO. uuuuuu.

The user number key in this record is outside the valid range. Supply the proper DD cards to bring up an alternate cluster for this user.

DIBIN603 I/O ERROR READING/WRITING ** DIRECTORY.

The **Directory will not be updated for this run. To obtain a **Directory, supply the proper DD cards to bring up an alternate cluster of data sets for all users.

DIBIN604 UNABLE TO CONVERT TTRN TO MBBCCHHR.

A serious problem has occurred with the data base. The systems programmer or analyst should be given the core dump produced.

DIBIN701 INSUFFICIENT CORE STORAGE ALLOCATED FOR POT'S.

Check to ensure that the CALL-OS job has been initiated in the desired task area. For MVT, ensure that the desired task area size was specified correctly. Increase the task area specifications and restart the job.

If Hierarchy 1 storage is being used, check the LCSRES parameter specification for a P to determine in which hierarchy the shortage has occurred.

DIBIN702 INSUFFICIENT CORE STORAGE ALLOCATED FOR 256-BYTE BUFFERS.

Check to ensure that the CALL-OS job has been initiated in the desired task area. For MVT, ensure that the desired task area size was specified correctly. Increase the task area specifications and restart the job.

If Hierarchy 1 storage is being used, check the LCSRES parameter specification for a B to determine in which hierarchy the shortage has occurred.

DIBIN703 INSUFFICIENT CORE STORAGE ALLOCATED FOR SORT BUFFER.

Check to ensure that the CALL-OS job has been initiated in the desired task area. For MVT, ensure that the desired task area size was specified correctly. Increase the task area specifications and restart the job.

If Hierarchy 1 storage is being used, check the LCSRES parameter specification for a S to determine in which hierarchy the shortage has occurred.

DIBIN704 INSUFFICIENT CORE STORAGE ALLOCATED FOR OVERLAY BUFFER.

Check to ensure that the CALL-OS job has been initiated in the desired task area. For MVT, ensure that the desired task area size was specified correctly. Increase the task area specifications and restart the job.

If Hierarchy 1 storage is being used, check the LCSRES parameter specification for an O to determine in which hierarchy the shortage has occurred.

DIBIN705 INSUFFICIENT CORE STORAGE ALLOCATED FOR UTT'S.

Check to ensure that the CALL-OS job has been initiated in the desired task area. For MVT, ensure that the desired task area size was specified correctly. Increase the task area specifications and restart the job.

If Hierarchy 1 storage is being used, check the LCSRES parameter specification for a U to determine in which hierarchy the shortage has occurred.

DIBIN706 INSUFFICIENT CORE STORAGE ALLOCATED FOR COMPILER AREA.

Check to ensure that the CALL-OS job has been initiated in the desired task area. For MVT, ensure that the desired task area size was specified correctly. Increase the task area specifications and restart the job.

If Hierarchy 1 storage is being used, check the LCSRES parameter specification for a C to determine in which hierarchy the shortage has occurred.

DIBIN707 COMMUNICATION AND COMMAND CONSOLE ASSIGNED TO SAME LINE.

The SYSCON and COMCON parameters point either to the same logical line number or to the default value of the other parameter. Correct the value and restart the job.

DIBIN708 COMMUNICATION CONSOLE SPECIFIED EXCEEDS NUMBER OF LINES OPENED.

The value specified in the COMCON parameter is greater than the highest relative line number specified in the current run. Correct the value and restart the job.

DIBIN709 SECOND COMMAND CONSOLE SPECIFIED EXCEEDS NUMBER OF LINES OPENED.

The value specified as the second command console in the SYSCON parameter in the EXEC card is greater than the highest relative line number specified in the current run. Correct the value and restart the job.

DIBIN710 SYSCON VALUE EXCEEDS NUMBER OF LINES OPENED.

The first SYSCON value specified is greater than the number of lines opened. Correct the value and restart the job.

DIBIN711 NOT ENOUGH WORK/SWAP CYLINDERS AVAILABLE FOR LINES OPENED.

There must be at least one cylinder of space available in the work/swap data sets for each line opened. Either include more SWAP DD cards or reduce the number of lines opened and restart the job.

DIBIN712 TRANSMISSION CONTROL UNIT NOT OPERATIONAL. CORRECT AND REPLY 'GO' OR REPLY 'CANCEL'.

The transmission control unit was found to be not operational after an I/O operation; possibly the unit has been turned off. Correct the problem and reply either 'GO' or 'CANCEL'.

DIBIN713 OS/360 SYSTEM DATE INVALID.

The OS/360 system date was set to a date prior to January 1, 1969. CALL-OS computes an internal date based on the OS/360 date and relative to January 1, 1969. CALL-OS cannot be initialized until the OS/360 date has been reset.

DIBIN714 IPBUFS VALUE EXCEEDS MAXIMUM.

The value specified for the IPBUFS parameter was greater than the maximum of 15 pots per line. A value of 15 is used.

DIBIN715 OPBUFS VALUE EXCEEDS MAXIMUM.

The value specified for the OPBUFS parameter was greater than the maximum of one per line. A value equal to the number of lines is used.

ONLINE ABEND CODES

In a relatively few instances, CALL-OS interrupts processing when a major error within the system is detected by the executive. Examples are not enough input buffers, the transient module M#LOG cannot be loaded as part of the enabling sequence, etc. When one of these conditions is detected, CALL-OS issues a user ABEND code and performs job termination processing. OS/360 issues the final termination message in the following format:

```
IEF450I jjj ABEND Uddd
```

where

jjj is the job name given in the CALL-OS startup deck
ddd is the user ABEND code issued by the executive

The message itself is explained in greater detail in the publication IBM System/360 Operating System: Messages and Codes. The possible user ABEND codes are given in numerical order along with the initiating module.

OS/360 Code	System Console Online ABENDs Explanation	User Action	Initiating Module
001	System has run out of pots (24-byte buffers).	Increase the number of pots allocated for each line with the IPBUFS parameter in the EXEC card of the startup deck.	S#GPOT
002	A CALL-OS routine has requested more than the 13 maximum allowable pots (24-byte buffers).	Contact the installation system programmer.	S#GPOT
003	Irrecoverable situation detected.	See CALL-OS communications console messages for clarification. Contact the installation system programmer	S#QEMSG
004	C#CPID has detected a CALL-OS timer expiration and level 4 (compilers and user programs) or level 5 (background) is not active.	Verify that batch programs are not generating invalid timer queuing elements or using unsupported timer code such as MFT timer exits. If batch programs are not doing the above, contact the installation system programmer.	C#CPID
005	C#CPID is attempting to put a level 4 (compilers and user programs) or level 5 (background) clock on when a previous clock on that level has not been taken off.	Verify that batch programs are not generating invalid timer queuing elements or using unsupported timer code such as MFT timer exits. If batch programs are not doing the above, contact the installation system programmer.	C#CPID

OS/360	System Console Online ABENDs		Initiating
Code	Explanation	User Action	Module
006	System has run out of pots (24-byte buffers).	Increase the number of pots allocated for each line with the IPBUFS parameter in the EXEC card of the startup deck.	C#CPID
007	System is unable to enable any command console.	Verify that terminal control unit and associated hardware is turned on and functioning properly.	M#CCDI

COBI MESSAGES

When COBI is used, certain messages appear on the OS/360 system operator's console. These messages may inform the operator of actions he is to perform, either on behalf of a user or to fulfill a request from a COBI module. In addition, these messages may also indicate error and/or status conditions of the COBI operations.

COBI messages will usually appear while CALL-OS is executing; however, if the COBI reader and/or writer program is executed offline, these messages may also appear when CALL-OS is not in operation. All COBI messages begin with an identifier of DIBxxnnn,

where

xx is a two-character letter code which indicates the source of the message: EX messages are from the executive, RD messages are from the COBI reader, and WT messages are from the COBI writer.

nnn is a three-digit number used to sequence the messages. The messages are listed in ascending sequence.

OS/360 System Console COBI Messages and Explanations

DIBEX001 tt:tt LINE nnn
userid: message

The terminal user may issue a NOTIFY CALL command at his terminal to transmit messages to the operator on the CALL-OS communications console. If the communications console is not initialized or fails during operation, a NOTIFY CALL message appears on the OS/360 system operator's console, prefixed by the identifier DIBEX001, and

tt:tt is the time of day

nnn is the logical line number

userid is the user number

message is the user-supplied message

The first line of this message may appear even when COBI is not used; in this case it precedes any CALL-OS message that was to have gone to the communications console but could not because the console was inoperative.

DIBEX002 userid: message

The terminal user may issue a NOTIFY OS command at his terminal to transmit messages to the operator on the OS/360 system operator's console. These messages are prefexed with DIBEX002, and

userid is the user number

message is the user-supplied message

DIBEX019 JOB xxxxxxxx CANCEL INITIATED BY COBI

The user issued a CANCEL command for one of his submitted jobs; the job to be cancelled has been read by a COBI reader but it has not completed execution. COBI issues a cancel request for the job with an SVC 34 and prints this message informing the operator that the request was initiated. The operator does not have to cancel the job.

DIBEX020 CANCEL COBI JOB xxxxxxxx WHEN DIBRDRx ENDS

When the COBI reader (either DIBRDRA or DIBRDRB) has completed reading the input data set, the operator must cancel the job whose name (xxxxxxxx) appears in the message. He does this with normal OS/360 cancellation procedures.

DIBEX021 nnnn JOBS ON SYSINx, START DIBRDRx

In this message, x is either A or B. The message notifies the operator of the number of jobs on the COBI input data set to be read and requests him to start the COBI reader. The operator must issue the appropriate start command from the OS/360 system operator's console.

This message may appear at any time during a COBI session if the manual mode of starting a reader is in effect. The operator may restore the automatic mode by issuing the *COBI RESET,AUTRDR=AUTO command from the command console.

If the ANYBATCH function is used during initialization and both input data sets are full, nnnn is replaced with blanks.

DIBEX022 PLEASE START REQUESTED DIBRDRx

The operator has not yet responded to a previous request to start a reader (where x is either A or B) for an input data set which is either full or has reached sufficient limits for switching. This previous request was issued with message DIBEX021, and indicates which reader is to be started.

The operator must issue the appropriate start command at the OS/360 system operator's console. If the input data set attached to COBI is not yet full, newly submitted jobs are placed on it; if this input data set is full, no more jobs may be submitted until the other input data set is available to be attached to COBI.

DIBRD003 userid job-name message

The prefix DIBRD003 is attached to all user-generated write-to-operator messages. These messages are submitted by the user as job control statements beginning with the character sequence //*. Generally, the messages are setup instructions to be followed by the operator when the job is executed. In this message,

userid	is the user number
job-name	is the name by which the job is known to OS/360
message	is the user-supplied message

DIBRD004 INITIALIZED FILE, NO JOBS PRESENT.

The operator has attempted to start the COBI reader before any jobs have been submitted. DIBRDR terminates.

DIBRD005 DIBRDRx INPUT DATA SET NOT CURRENTLY AVAILABLE

The operator has attempted to start either DIBRDRA or DIBRDRB, but CALL-OS has not made the required data set available. It may be, for example, that CALL-OS is currently constructing the data set, in which case, it will provide notification to the operator when the data set is complete. The reader is terminated and must be restarted when the input data set is available.

DIBRD006 EOF PRIOR TO LAST WTO MESSAGE

There are additional WTO messages for the current job, but an end-of-file condition has been encountered. This message warns the operator that he has not received all WTO operator messages for the job. Generally, an OS/360 or hardware-related problem has occurred and the system programmer should be notified. Execution of the COBI reader program continues with reading of the next job.

DIBRD007 UNABLE TO OPEN INPUT DATA SET, DIBRDR ENDED

Execution of the COBI reader program (DIBRDR) cannot be successfully initiated. No jobs can be processed. Generally, an OS/360 or hardware-related problem has occurred either during open or during the first read operation on the data set. The installation system programmer should be notified.

DIBRD008 UNABLE TO READ WTO MESSAGES

Execution of the COBI reader program has been initiated, and the required input data set has been opened. There are some WTO messages for the first jobs in the input data set, but the system is unable to read (and, subsequently, print) those messages. Generally, an OS/360 or hardware-related problem has occurred.

DIBRD009 UNABLE TO COMPLETE WTO MESSAGES

Execution of the COBI reader program has been initiated, the required input data set has been opened, and WTO messages for one or more jobs have been processed successfully. Now, however, the system is unable to process WTO messages. Generally, an OS/360 or hardware-related problem has occurred.

DIBRD010 EOF SET, DIBRDR TERMINATED

Execution of the COBI reader program has been initiated, and the required input data set has been opened, but there are no jobs to be processed. A program error has occurred and the installation system programmer should be notified. Execution is terminated.

DIBRD011 INCORRECT BLOCKSIZE, DIBRDR TERMINATED

The system has attempted to calculate the number of records per block, by dividing the block size indicated in the data control block (DCB) by 80, which is the number of characters per record. However, the result is not a positive number. Execution is terminated. The user should check the BLKSIZE subparameter in the DCB parameter for the data set.

DIBRD012 ABNORMAL READER TERMINATION, JOBNAMES FOLLOW

Execution of the COBI reader program has been initiated, and the required input data set has been opened. The COBI reader program attaches the OS/360 reader program to read the appropriate data set. For some reason, the OS/360 reader program cannot continue processing and it terminates abnormally. This message is followed by a list of the names of all the jobs in the data set being read. The list is followed by message DIBRD015, which instructs the operator to restart the reader.

DIBRD013 UNABLE TO READ JOBNAMES,DIBRDRx TERMINATED

Execution of the COBI reader program either has been initiated with the parameter $$$$ (to display job names) or was initiated normally but the OS/360 reader terminated abnormally. There are some jobs in the input data set, but the job names cannot be read. Generally, an OS/360 or hardware-related problem has occurred. The COBI reader program terminates. In this message, the x is replaced by either A or B.

DIBRD014 UNABLE TO COMPLETE JOBNAMES

Execution of the COBI reader program either has been initiated with the parameter $$$$ (to display job names) or was initiated normally but the OS/360 reader terminated abnormally. One or more jobs were read, but the reader could not finish reading the names of the other jobs in the data set. Generally, an OS/360 or hardware-related problem has occurred. The COBI reader program terminates.

DIBRD015 COMPARE Q AND JOBNAME LIST, RESTART DIBRDRx,,,(NEXT JOBNAME)

The message follows the list of job names printed after message DIBRD012. The operator should compare the names of the jobs in the job queue against the list to determine the next COBI job to be processed. He then should start the specified reader at the next job. In this message, the x is replaced by either A or B.

DIBRD016 DID NOT LOCATE FILE, RETRY

For some reason, the COBI reader could not read the first record of an input data set. The operator must reissue the previous start command, specifying the appropriate reader (DIBRDRA or DIBRDRB). If this message appears again, it indicates a possible hardware error and the installation system programmer should be notified.

DIBRD017 FILE PREVIOUSLY READ. REPLY 'CANCEL' OR 'CONT'.

The COBI reader detects that the input data set has already been read. The operator may either cancel reader operation or continue processing by rereading the data set. If he decides to continue, message DIBRD018 is issued.

DIBRD018 REPLY 'YES' TO INCLUDE WTO MESSAGES; ELSE 'NO'.

This message appears after the operator has replied 'CONT' to message DIBRD017. The replies have the following effect:

1. A reply of 'YES' causes the jobs to be reread and all WTO messages associated with those jobs are reissued.

2. A reply of 'NO' causes the jobs to be reread but the WTO messages associated with the jobs are ignored.

DIBWT041 COBI CLASS NOT IN PARM OF EXEC CARD

The CBCLASS parameter was omitted from the EXEC statement in the DIBWTR procedure. This parameter specifies the COBI output class; this class is intercepted by DIBWTR for processing. The procedure must be corrected before DIBWTR can be executed.

DIBWT042 OS CLASS NOT IN PARM OF EXEC CARD

The OSCLASS parameter was omitted from the EXEC statement in the DIBWTR procedure. This parameter specifies the OS/360 output class to which the COBI class is transferred after processing by DIBWTR; this OS/360 class is printed by an output writer assigned to the class. The procedure must be corrected before DIBWTR can be executed.

DIBWT043 DISK ERROR READING VOLID TABLE

A permanent I/O error occurred while trying to read the volume identification table from the COBI JCL data set. This condition could indicate a hardware problem; contact the installation system programmer.

DIBWT044 VOLID TABLE IDENTIFIER MISSING

The volume identification table was read successfully from the COBI JCL data set; however, the expected identifier CBVD was not found in the first record. Contact the installation system programmer.

DIBWT045 DISK ERROR READING MASTER QCR CONTROL RECORD

A permanent I/O error occurred while reading the master queue control record from SYS1.SYSJOBQE. Contact the installation system programmer.

DIBWT046 DISK ERROR READING COBI CLASS QCR RECORD

A permanent I/O error occurred while reading the queue control record for the COBI output class. Contact the installation system programmer.

DIBWT047 DISK ERROR READING LOGICAL TRACK HEADER

A permanent I/O error occurred while reading a logical track header from the COBI output class. Contact the installation system programmer.

DIBWT048 DISK ERROR READING LOGICAL TRACK

A permanent I/O error occurred while reading a logical track from the COBI output class. Contact the installation system programmer.

DIBWT049 **** RESET 1ST JOB IN COBI SYSOUT CLASS TO OS CLASS

For some reason, DIBWTR cannot process the first job in the COBI output class. The operator should display the names of the jobs in the queue for the class and reset the first job to the OS/360 output class.

DIBWT050 TIME INTERVAL IN PARM NOT VALID

Either the time interval was not specified in the ITIME parameter for DIBWTR or, if present, the interval is either zero, or greater than 9999, or contains other than numeric characters. Correct the parameter field and restart the job.

DIBWT051 DISK ERROR READING COBI INDEX RECORD

A permanent I/O error occurred while reading the COBI index record. Contact the system installation programmer.

DIBWT052 NO MORE COBI JCL SPACE AVAILABLE

The COBI writer has attempted to write the JCL for a job in the COBI JCL data set but there is no more space available in the data set. The job is reassigned to the OS/360 message class for processing by an output writer.

DIBWT053 DISK ERROR SEARCHING FOR AVAILABLE JCL SPACE

A permanent I/O error occurred while the COBI writer was searching for space in which to write the JCL for a COBI job. The job is reassigned to the OS/360 message class for processing by an output writer. This error may indicate a hardware problem and the installation system programmer should be contacted.

DIBWT054 FAILURE TO OPEN DIBWTR DATA SETS

All or one of the following data sets cannot be opened: CBJCL, CBNDX, or SYS1.SYSJOBQE. Since these data sets are defined in the DIBWTR cataloged procedure, contact the installation system programmer.

DIBWT055 DISK ERROR WRITING COBI INDEX RECORD

A permanent I/O error occurred while writing the COBI index record. The job is reassigned to the OS/360 message class for processing by an output writer. This error may indicate a hardware problem and the installation system programmer should be contacted.

DIBWT056 VOL SER JCL MESSAGE NOT FOUND FOR DSN=DIB...

Ordinarily, one JCL allocation message appears for each COBI scannable data set. This allocation message indicates the volume serial number of the volume on which the data set was written. The preceding error message indicates that the allocation message did not appear, which indicates an error in the OS/360 job queue. The installation system programmer should be contacted.

DIBWT057 DISK ERROR WRITING COBI JCL RECORD

A permanent I/O error occurred while writing the COBI JCL record for a job. The job is reassigned to the OS/360 message class for processing. This error may indicate a hardware problem and the installation system programmer should be contacted.

DIBWT058 UNABLE TO FIND DIBWTR CSCB

The DIBWTR task, started at the OS/360 system operator's console, cannot continue processing because the command scheduling control block (CSCB) could not be found. This makes it impossible to terminate the DIBWTR with the STOP command. Contact the installation system programmer.

DIBWTR ABEND CODES

When COBI is used, the COBI writer (DIBWTR) is executed when CALL-OS is no longer in operation. During its execution, DIBWTR may issue ABEND codes when it can no longer continue. OS/360 issues the final termination message in the following format:

```
IEF450I  DIBWTR  ABEND Uddd
```

where

ddd   is the ABEND code issued by DIBWTR

The message itself is explained in greater detail in the publication IBM System/360 Operating System: Messages and Codes. The possible ABEND codes are given in numerical order along with an explanation.

OS/360 System Console DIBWTR ABENDs

Code	Explanation
001	An error has been detected in TTR translation.
003	One of the data sets required for DIBWTR operation cannot be opened; these are the COBI index, JCL, and OS/360 job quque data sets.
321	An incorrectly compressed job queue record has been detected.

UTILITY ABEND CODES

In relatively few instances, the CALL-OS utilities interrupt processing when a major error is detected. Examples are inability to open a required data set. When one of these error conditions is detected, the utility issues a user ABEND code and terminates. OS/360 issues the final termination message in the following format:

```
    IEF450I jjj ABEND Uddd
```

where

jjj   is the job name

ddd   is the user ABEND code

The message itself is explained in greater detail in the publication IBM System/360 Operating System: Messages and Codes. The possible ABEND codes are given in numerical order along with the initiating module.

OS/360 Code	System Console Utility ABENDs Explanation	Initiating Module
111	Bad open of the control card input data set. The SYSIN DD card is probably missing.	UTILX
112	Bad open of the printer output data set. The SYSPRINT DD card is probably missing.	UTILX
113	Bad open of the punch card output data set. The SYSPUNCH DD card is probably missing.	UTILX
114	Bad open of the index data set. The INDEX DD card is probably missing.	UTILX
115	End-of-file encountered when reading first record in the index data set. The index has probably not been formatted using U#UTIL3.	UTILX
200	Bad open of the SYSPRINT data set. The SYSPRINT DD statement is probably missing. This ABEND is issued when an error is encountered in the parameter field and the attempt to write a message in the SYSPRINT data set failed.	U#UTIL2
505	The SYSPRINT data set could not be opened. The SYSPRINT DD statement is probably missing.	U#5INIT U#5RINIT U#5CBXPN U#5PURGE
507	Error detected during a COBI data set operation.	U#5INIT U#5CBXPN U#5PURGE
601	The SYSPRINT data set could not be opened. The SYSPRINT DD statement is probably missing.	DIBCONPR
701	The index data set could not be opened. The INDEX DD card is probably missing.	U#UTIL3
750	The attempt to open the SYSPRINT data set for output was unsuccessful. U#UTIL1 terminates without processing any DD cards for index data sets.	U#UTIL1

OS/360 System Console Utility ABENDs

Code	Explanation	Initiating Module
760	The attempt to open the index data set for update was unsuccessful. U#UTIL1 terminates without processing any DD cards for INDEX data sets.	U#UTIL1
770	An end-of-file condition was detected on the attempt to read the first (and only) record in the index data set; that is, the data set is empty. U#UTIL1 terminates without processing any DD cards for index data sets. U#UTIL3 should be used to format the index to a one-record single track of X'FF's.	U#UTIL1
800	An end-of-file condition was detected on the attempt to read the first (and only) record in the index data set; that is, the data set is empty.	DIBINIT
801	Unable to open the SYSIN data set.	DIBINIT
802	Unable to open the SYSPRINT data set.	DIBINIT
803	Unable to open the CALL-OS index.	DIBINIT
891	DIBDISKA was entered, but no preceding call to DIBDISK for this data set has been given.	DIBDISK
892	A second call was made to DIBDISK to read or write a data set before the required call to DIBDISKA for that data set was given.	DIBDISK
893	An invalid parameter was specified in a call to DIBDISK. For example, an invalid relative address.	DIBDISK

UTILITY MESSAGES

These error messages may be printed at the OS/360 system operator's console during execution of a CALL-OS utility program. The messages are in alphabetical order, either according to the message identifier DIBxxxxx or according to the first word of text if no identifier is present.

OS/360 System Console Utility Messages and Explanations

DIBUT201 MVT/MFT VERSION CALL-OS BEING BUILT. IPL NEW NUCLEUS AT COMPLETION.

This message is issued by U#UTIL2 when it begins building CALL-OS. The operator is to IPL the new OS/360 nucleus created during system build.

DIBUT505 BAD OPEN FOR SYSPRINT, SEE ABEND 505

The SYSPRINT data set could not be opened by one of the COBI utilities (U#5INIT, U#5RINIT, U#5CBXPN, or U#5PURGE). The ABEND message with the code 505 indicates the name of the job which could not open the data set. If no SYSPRINT DD statement was supplied, add one to the deck. If the SYSPRINT DD statement was supplied, ensure that the information on it is correct. If the DD statement is present and is correct, notify the installation system programmer immediately.

DIBUT510 NO SCANXX DD CARDS FOUND IN TIOT TABLE. SHOULD U#5PURGE BE CONTINUED? REPLY 'YES' OR 'NO'

Part of the purge process is the scratching of any scannable output data sets associated with unfinished COBI jobs. A reply of 'YES' indicates that the process should continue and that scannable output data sets need not be present. A reply of 'NO' indicates that SCANxx DD statements must be supplied and processing is terminated after message DIBUT531 is written on the system printer.

DIBUT511 CBSYSINx CONTAINS INPUT DATA. SHOULD xxxxxxxx BE CONTINUED? REPLY 'YES' OR 'NO'

Ordinarily, the COBI input data sets should have a COBI EOF record as the first record. This message indicates that either new space was not allocated to the data sets (for U#5RINIT) or the specified data set has not yet been read by the COBI reader (for either U#5RINIT or U#5PURGE). If the operator wishes processing to continue and the contents of the input data set ignored, he replies 'YES'. If he wishes processing to terminate so that, for example, the input data set may be later processed by COBI, he replies 'NO'.

DIBUT601 OPEN FAILED FOR SYSPRINT, SEE ABEND 601

The SYSPRINT data set could not be opened by the DIBCONPR utility. The ABEND message with the code 601 indicates the name of the job which could not open the data set. If no SYSPRINT DD statement was supplied, add one to the deck. If the SYSPRINT DD statement was supplied, ensure that the information on it is correct. If the DD statement is present and correct, notify the installation system programmer.

*** PARM FIELD = xxxxxxxx ***CANNOT DECIPHER*** ***REPLY 'COMPILER', 'SYSGROUP', 'USRGROUP', 'WORKSWAP' OR 'OVERLAY'***

This message is issued by U#UTIL1. It indicates that the function specified in the PARM field on the EXEC card has been omitted or misspelled.

CALL-OS COMMAND CONSOLE MESSAGES

The command console receives messages in response to operator commands entered at the console during online operation of the system. In addition, the command console may receive messages that also appear on the user's terminal. This section only lists the messages which appear in response to an operator command. If a message appears at the command console that is not in this list, the operator should consult the publication CALL-OS Terminal Operations Manual for a description of the message. The responsible system programmer or analyst should review all command console messages daily since some of these messages may indicate system difficulties which should be corrected.

The messages are listed in alphabetical order. Those messages which begin with variable information (for example, a user number) are alphabetized according to the second word in the message. For example, the message IBM406 NOT CURRENTLY ASSIGNED is alphabetized according to the word NOT.

Command Console Messages and Explanations

ALLOCATICN RECORD NOT IN PROPER GROUP. PROCESSING HALTED.

During the *OFF sequence, the user group range field of the allocation record does not equal the user group range of the UGT (user group table to data set translation table) entry.

ALTERNATE SYSIN NOT AVAILABLE

The *COBI ANYBATCH command has been issued and there are jobs available to be given to OS/360; however, the alternate COBI input data set is not available for use by COBI.

DELETED

The operator has caused the current line being entered to be deleted.

DCB NUMBER IS GREATER THAN MAXIMUM NUMBER

During the *OFF sequence, the DCB number being processed is beyond the upper limit of DCB numbers for the current version of CALL-OS. This is a system problem and the installation programmer should be notified.

DS NAME SYSINx HAS NO SUBMITTED JOB

Data set status of either SYSINA or SYSINB was requested but there are no jobs associated with the data set. Either a reader has not been started by the operator or no jobs have been submitted by terminal users. Processing of the command is terminated.

END OF STATISTICAL REPORT

*REPORT request processing is finished.

ENTER

The operator has entered *DATE, *MESSAGE, a form of *TELL, or *WARN; this message signals him to supply the message.

EXPIRATION DATE HAS NOT BEEN REACHED

The operator has issued a *COBI SCRATCH command for a data set and/or a job. The scratch request cannot be fulfilled because the expiration date has not been reached.

FUNCTION NOT AVAILABLE

The NOCOBI parameter was specified in the startupdeck for CALL-OS and the operator has issued any form of the *COBI command.

INVALID CHARACTERS IN USER NUMBER

The user number supplied with the *VALIDATE command contains either nonalphabetics in the first three characters or nonnumerics in the second three characters.

INVALID COMMAND

A parameter specified for the *TELL command does not start with any of the following:

1. OP
2. E-MSG
3. D-MSG
4. An alphabetic character for a user number (A-Z)
5. A numeric character for a line number

INVALID LINE NUMBER(S)

For *TELL, a line number was specified for a line that has not been enabled.

For *ENABLE or *DISABLE, one of the following applies to the line number(s) specified:

1. More than two line numbers are specified.
2. A line number is not numeric.
3. A line number is less than one or greater than the highest logical line number in the system.

INVALID LINE NUMBER--COMM. CONSOLE

A *TELL request for communications console specifies the line number. The operator must reenter the command and specify OP.

INVALID PARAMETER

For *STATUS, this message is printed if neither a line number nor a user number is entered with the command.

The operator has entered *VALIDATE and has not supplied a password. The user number indicates this is not the creation of a *Directory.

INVALID TERMINAL NUMBER

This message is printed in response to the *STATUS command if the line number parameter specifies a line that has not been enabled.

INVALID USER NUMBER

The *TELL command specifies a user number that has not been validated.

INVALID USER NUMBER RANGE IN EQUIV RECORD

During the *VALIDATE process, the user number range contained in the equivalency record does not equal that in the UGT (user group to data set translation table).

aaannn IS ALREADY ASSIGNED

The *VALIDATE command has been issued for a user number (aaannn) that has already been validated. The operator must reissue the command and specify the C parameter; only the password of the user is changed.

JOB ID #nnnnn HAS NO COBI INDEX RECORD

Either a job number (#nnnnn) was specified in a *COBI command but no COBI index record exists for the job or status was requested for all jobs but no jobs exist in the COBI index (in this case, JOB ID..... appears in the message). Processing continues with the next parameter, if any.

JOB ID #nnnnn HAS NO SCANNABLE OUTPUT

A request for data set status was issued and either the indicated job has not completed execution, or no scannable data sets were specified when the indicated job was submitted and no JCL errors were detected. In either case, the message indicates that no scannable output exists for the job at this time.

JOB ID #nnnnn . . INVALID JOB NUMBER

The job number (#nnnnn) specified is either less than or equal to one, or greater than the highest permissible job number. Processing continues with the next parameter, if any.

xxxx JOBS ON SYSIN, AUTO DIBRDR STARTED

The operator has issued the *COBI ANYBATCH command and the specified number of jobs are to be made available to OS/360 for processing. A reader was started for these jobs automatically.

LINE xxx INACTIVE

The *TELL command specifies the line number of an inactive line.

LINE xxx INACTIVE -- MESSAGE CANCELLED

This message is printed if the receiving terminal signs off before *TELL message input is completed.

M#CCTE ERROR, UTT ADDRESS NOT EQUAL TO G#TEUTT

An error is detected when a terminal is ready to receive a *TELL message. The UTT address of a line number for *TELL in GTAB is different from the address of the UTT of an addressee. This is a system problem and the installation programmer should be notified.

MESSAGE FOR LINE xxx BEING SENT

A *TELL message saved for a terminal is in the process of being sent by the CPU; xxx is the line number of the terminal.

MESSAGE FOR LINE xxx CANCELLED

A message that has been entered for the *TELL command is cancelled by a command console request.

MESSAGE IS TOO LONG

The operator has entered a message that exceeds 74 printable characters plus carrier return. Repeat the command (*MESSAGE, *TELL, or *WARN) and in response to ENTER, issue a shorter message.

MESSAGE PENDING FOR LINE xxx ENTER

Response to a *TELL request if there is a queued *TELL message. The requester can either wait until the pending message has been sent (depress RETURN) or overlay the message text by entering another message.

MORE THAN 24 CHARACTERS. RETYPE IT

The operator has attempted to enter a *DATE message containing more than 24 characters. The operator must enter a shorter message.

NO DCBS IN TABLE. ALLOCATION RECORDS NOT PROCESSED

During the *OFF sequence, the UGT (user group to data set translation table) contains the table delimiter where the first entry should be. This is a system problem and the installation system programmer should be notified.

NO FULL TRACK AVAILABLE

A request has been made for a full track to be used to create a new (additional) equivalency class record. Indicates no full tracks available for this user group.

NO HALF TRACK AVAILABLE

A request has been made for disk space to be used to create a new catalog or directory file, and the parameter returned indicates no disk space available.

NO JOB AVAILABLE FOR BATCH

The operator has issued the *COBI ANYBATCH command but there are no jobs in the input data set attached to COBI.

NO PENDING MESSAGE

Response to a display (D-MSG) or erase (E-MSG) request if there is no message pending.

NO SUCH USER

The operator has entered *CANCEL and the user number supplied is not represented in the equivalency file.

aaannn NOT CURRENTLY ASSIGNED

The *VALIDATE command has been issued requesting that the password be changed for a currently validated user number (aaannn). However, the user number is not assigned; the *VALIDATE command must be reissued without the C parameter.

NOT SCRATCHED: xxxx c, xxxx c,...

The operator has requested the scratching of a data set(s), but his request cannot be satisfied because of conditions encountered during processing. Each xxxx is an entry of the form nPmm or Unnn identifying a data set that cannot be scratched. Each c is a code indicating why the specified data set cannot be scratched; possible code values and their meanings are:

1 - Background using data set
2 - System problem
3 - Data set not on volume indicated
4 - Data set not in COBI index
5 - Volume not in volume identification table
6 - System problem of an I/O nature
7 - Volume not available

PARAMETER,xxxxxxxxxxxx, INVALID ENTRY

An input parameter (xxxxxxxxxxxx) for the *COBI command cannot be processed successfully. This message appears under the following conditions:

1. The user's job number (of the form #nnnnn) contains more than five digits, only the # character, or other than numeric characters.

2. The user number (of the form aaannn) is not equal to six characters, or the first three characters contain a nonalphabetic character, or the last three characters contain a nonnumeric character (SYSLIB is the only exception), or the user number is not found in the equivalency record as a validated user number.

3. For DSSTATUS and JOBSTATUS, job numbers and user numbers are specified on the same command.

Processing continues with the next parameter, if any.

READY

This is the normal system response when a requested operation is completed successfully.

RECORD READ NOT EQUIVALENCY CLASS

The first four bytes of the key field of a record read as part of an equivalency file containing something other than EQU.

RE-ENTER COMMAND, LINE NUMBERS REQUIRED FOR DISABLE

Parameters must be supplied with the *DISABLE command.

SEEK ADDRESS DOES NOT CORRESPOND TO CURRENT DCB

During the *OFF sequence, the relative data set number of the allocation record read does not equal the relative data set number being processed. This is a system problem and the installation system programmer should be notified.

START DIBRDRx

In this message, x is either A or B. The message notifies the operator that a reader must be started to process COBI jobs. The operator must start the appropriate reader from the OS/360 system operator's console.

This message may appear at any time during a COBI session if the manual mode of starting the reader is in effect. The operator may cause the automatic mode to be in effect by issuing the command *COBI RESET,AUTRDR=AUTO at the command console.

SYNTAX ERROR

The operator has entered a form of the *COBI command which contains a syntax error. He should reenter the command correctly.

SYSPRINT DATA SET UNAVAILABLE/[REPORT] REQUEST TERMINATED

The operator has specified *REPORT or *COBI-P and the SYSPRINT DD card was not included in the startup deck. The request is terminated. For *COBI-P, the operator may reissue the request by specifying *COBI.

SYSTEM OFF AT -- tt:tt

This message signals completion of the *OFF process.

SYSTEM PROBLEM, RETRY

An unrecoverable disk I/O error has occurred during an attempt to respond to an operator request; the operator should reenter his request.

UNABLE TO READ EQUIVALENCY CLASS

During the *VALIDATE process, an irrecoverable disk error occurred while attempting to read an equivalency class record.

UNABLE TO WRITE HALF TRACK

During the *VALIDATE process, an irrecoverable disk error occurred while attempting to write a catalog or directory record.

UNRECOVERABLE I/O (PROGRAM) ERROR INDICATION READING (WRITING)
ALLOCATION FILE. RECORD NOT UPDATED

During the *OFF sequence, an error indication was found at the completion of a disk I/O operation involving the allocation record.

USER GROUP DATA SET(S) NOT ACTIVE

The user number entered on a *VALIDATE command does not fall within the range of an entry of the UGT (user group to data set translation table).

USER ID aaannn HAS NO COBI INDEX RECORD

The catalog for user number aaannn has been completely scanned and either no COBI job entries were found or the COBI job entries found were invalid. Processing continues with the next status request parameter, if any.

USER NUMBER NOT FOUND

This message is printed if the user number parameter specified on a *STATUS command is not equal to a valid user number.

nn USERS AT -- tt:tt

The operator has entered *USERS. This is the standard response, where

nn is the number of active users

tt:tt is the time of day

CALL-OS COMMUNICATIONS CONSOLE MESSAGES

Each operations center should be equipped with a communications (print only) console for online system communication with the operator. The severity of the message governs the operator's corrective action. The responsible systems programmer or analyst should review the communications console output daily since some of these messages may indicate system difficulties which should be corrected. The messages are listed according to function as follows:

- System messages issued by the executive
- User-related error codes issued by the executive
- Disk I/O error messages
- Compiler-related error messages

When printed, each message is preceded by a heading line of the following format:

```
tt:tt   LINE nn
```

where

tt:tt is the time of day

nn is the line number

The messages themselves are described in the following sections.

When COBI is used, the terminal user may transmit messages and instructions to the communications console. These messages appear on the communications console in the following format:

```
tt:tt   LINE   nn
aaannn: message
```

where

tt:tt and nn have the same meaning as described previously

aaannn is the user number

message is the user-specified message

In those situations where a communications console has not been provided or where one has become inoperative, all messages intended for this console are sent to the OS/360 system operator's console with the following heading line:

```
DIBEX001 tt:tt   LINE   nn
```

where

tt:tt and nn have the same meaning as described previously

SYSTEM MESSAGES ISSUED BY THE EXECUTIVE

Messages are issued by the executive to communicate system status by identifying problems as they occur. Operator invervention will be noted when necessary.

Communications Console System Messages and Explanations — Initiating Module

AT tt:tt, LINE nn — M#ERMSG
COMMUNICATIONS CONSOLE IS ACTIVE.
SYSTEM ENABLED AT hh:mm.

This message is transmitted to the communications console whenever it is brought up and connected to the system either for the first time or from any on-hook condition.

tt:tt is the time of day when the communications console is connected to the system.

nn is the line number of the communications console.

hh:mm is the time of day when one or more command consoles are enabled and the system is initialized.

CANNOT LOAD COMPILER cccccccc — S#LDSP

An irrecoverable disk error occurred while reading compiler cccccccc. May be due to system build procedure error.

CENTRAL CYLINDER DISK ERROR ON SOURCE READ — M#RDSO

An irrecoverable disk error occurred while reading the user's work area.

CENTRAL CYLINDER DISK ERROR ON SOURCE WRITE — M#WRSO

An irrecoverable disk error occurred while writing the user's current sorted source program to his work area.

CENTRAL CYLINDER DISK ERROR ON SWAP READ — M#ISWP

An irrecoverable disk I/O error was detected while trying to read the user's swap area.

CENTRAL CYLINDER DISK ERROR ON SWAP WRITE — M#OSWP

An irrecoverable disk error has occurred while trying to swap out a user program.

CHAN OR 2702/3 BUSY FOR 20 SIOS. USER DISABLED — I#NEAPP

A condition code from a start I/O to a terminal I/O indicates one of the following:

1. The terminal is not available.

2. The 2702/3 TCU shows a busy status.

3. A busy channel status is given in the 20 start I/O attempts.

The terminal is disabled.

COMMAND AND COMMUNICATIONS CONSOLES ENABLED
YEAR = xxxx, DAY = yyy — M#CCDI

CALL-OS has been activated and both command and communications consoles are enabled by the system. If a communications console is unavailable at this time, the OS/360 system operator's console will receive this message, and messages addressed to the communications console will continue to be sent to the OS/360 system operator's console unless a communications console is made available.

xxxx = year (for example, 1970)
yyy = day (for example, 365 for December 31st)

DISK READ ERROR FOR SAVED PROGRAM — M#LDRD

An irrecoverable disk I/O error was detected while trying to read the user's program from save storage.

EQUIVALENCY CLASS DISK ERROR AT SIGN-ON TIME — M#LOG

An irrecoverable disk error has occurred while trying to read the equivalency class file.

EQUIVALENCY CLASS DISK ERROR DURING SIGN-OFF — M#LOG

An irrecoverable disk error has occurred while trying to read or write the user's equivalency class file.

ERROR MESSAGE QUEUE EXHAUSTED. MESSAGE(S) LOST — S#QEMSG

The number of messages being routed to the communications console has exceeded the capacity of the queue.

ERROR RATIO EXCEEDED — I#NEAPP

Ten errors have been detected in the last 500 I/O operations on this line. The user is disconnected but the line is not taken out of service. An error recording has been made on SYS1.LOGREC. The operator may want to take the line out of service with a *DISABLE command if the error persists.

ERROR THRESHOLD EXCEEDED — I#NEAPP

Excessive consecutive error conditions have been detected on this line. An error recording has been made on SYS1.LOGREC. The user is disconnected but the line is not taken out of service. The operator may want to take the line out of service with a *DISABLE command if the error persists.

xxxx HAS ATTEMPTED MOD CALL WITHOUT RELEASING OVLY BUFFER M#OVLY

The module represented by xxxx has failed to relinquish the overlay buffer. This is a CALL-OS system problem.

HEX xx, ERR RECORDED, LINE LEFT IN SERVICE, BUSY TEL LINE IF ERROR PERSISTS I#NEAPP

An error was recorded for the logical line (nnn, given in message header) associated with the indicated device address (xx). The current user has been successfully dropped by the system and the specific error has been recorded in SYS1.LOGREC. The statistical data recorded for the line may contain valuable information, so this information should be consulted on the error record printout. The error record is identified by time stamp (tt:tt, given in message header) and device address (xx).

This message usually precedes one of the following two messages:

ERROR RATIO EXCEEDED
ERROR THRESHOLD EXCEEDED

HEX xx, LINE OUT OF SERVICE, ERROR RECORDED, ***BUSY OUT TELEPHONE*** I#NEAPP

The logical line (nnn, given in message header) associated with the indicated device address (xx) is taken out of service due to a hardware or software system problem; an error recording has been made on SYS1.LOGREC. The operator must either busy-out the telephone associated with this line or reenable the line with the *ENABLE command. To reenable, he uses the logical line number specified in the message header.

NO TRANSLATION TABLE. DISABLED. I#NEAPP

This message is printed on the communications console when a user terminal or command console receives an INVALID TYPE DISABLED message due to lack of translation table. A header line precedes this message giving the logical line number of the affected terminal.

A 2741 terminal on the indicated line is disabled because the translation table for this terminal type is not core resident.

The major causes for this are the following:

1. The 2741 correspondence terminal is not specified in any of the terminal DD cards in the system startup deck.

2. The correspondence translation table is not specified in the RESMODS entry.

```
SYSTEM ERROR, *** BUSY OUT DATA SET ***                     M#CCDI
ERROR IN MODULE M#CCDI, CODE 2
```

When the system is in the process of disabling or aborting a terminal, an *ENABLE command is received. The above error message is output and the terminal is disabled.

```
TYPE LOG OR RETURN KEY/(garbled data)                       I#NEAPP
(garbled data)/TYPE LOG OR RETURN KEY
```

One of the above two messages is printed if the 2741 is used as the communications console and its operator fails, in the course of the sign-on procedure, to input either:

1. LOGON/LOG and RETURN, or

2. RETURN for a default 2741 type.

If the terminal is a 2741 correspondence, the first message is printed; if it is 2741 EBCD, the second message is printed.

Unlike a user terminal and command console, the communications console is expected to transmit either LOGON/LOG or RETURN to the system before it enters the RECEIVE mode.

```
UNABLE TO READ OVERLAY MODULE -- L#STAT=xx                  M#OVLY
```

A disk error has occurred while reading a module in the xx sequence; see Appendix A.

```
USER aaannn x-RD, x-WR, x-IR, x-BUS, x-EQ, x-OR, x-LD, x-TO    I#NEAPP
```

This message identifies the user number (aaannn) and displays the IOS error counter contents for the line, where x is the number of times one of the following conditions occurred:

RD	Read error
WR	Write error
IR	Intervention required
BUS	Bus out check
EQ	Equipment check
OR	Overrun
LD	Lost data
TO	Time out

This information appears immediately after any of the following messages:

```
ERROR RATIO EXCEEDED
ERROR THRESHOLD EXCEEDED
HEX xx, ERROR RECORDED, LINE LEFT IN SERVICE ...
HEX xx, LINE OUT OF SERVICE ...
```

USER-RELATED ERROR CODES ISSUED BY CALL-OS EXECUTIVE

Certain conditions detected within CALL-OS cause a coded message to be printed on the communications console. All such messages should be brought to the attention of the installation system programmer. The format of this type of message depends on whether the module is resident or nonresident, that is, executing from the overlay buffer. The format for resident modules is:

ERROR IN MODULE mmmmm, CODE xx

The format for nonresident modules is:

ERROR IN MODULE NO. hhh, CODE xx

where

mmmmm is the initiating module name

hhh is the module number in hexadecimal format (see Appendix B)

xx is the error code

The error codes are listed below along with the initiating module.

Communications Module	Console Code	User-Related Error Codes Explanation
I#NEAPP	1	Zero 24-byte buffer link field found in input buffer string during character deletion. (Buffers are lost from system on recovery.)
	2	Error on halt I/O command
	3	Terminal wake-up vector is zero.
	4	Illegal channel state (L#TCHST)
	5	200 successive extraneous interrupts from an out-of-service or non-CALL/360-OS device
	6	32 successive breaks
	8	Lost data while reading paper tape
	9	Sense byte is zero after a sense operation.
	10	Start I/O error
	11	Missed PCI interrupt due to system load
	12	System too slow to issue halt I/O or break test
	13	Illegal I/O state in start I/O appendage.
	14	Halt I/O error after a prepare command.
	15	Three consecutive executions of illegal CCW
	16	No carrier return character found before the last TTY input data in the pot is translated

Communications	Console	User-Related Error Codes
Module	Code	Explanation
I#PCIAPP	1	Channel state incorrect on PCI interrupt
	2	The number of edited input buffers ready for disk is greater than the total number of input buffers assigned to the line.
M#ABOR	1	The user's task is not abortable and the abort bit is already set.
M#ACCT	1	An error has occurred while reading a disk equivalency record.
	2	An error has occurred during the equivalency search. No match was found for the user number in the equivalency file.
	3	An error has occurred while writing a disk equivalency record.
M#CANCL	2	Unable to locate catalog record
	3	Job number greater than size of index string
	4	Invalid return from dequeue of job name
	5	Unable to locate header file for COBI input data set
	6	Invalid return from dequeue of job name (other data set)
	7	Invalid return from dequeue of input data set
M#CAT	1	This error message is issued when a user program is opening a file and the additional core requirement specified for large program file buffers cannot be satisfied. The terminal user receives a message indicating that his program has exceeded memory.
M#CBIO	1	The calling module has passed an RTTR (relative record number) less than one to M#CBIO.
	2	A TTR given to the OS/360 convert routine has resulted in a nonzero condition in register 15; the I/O cannot proceed.
M#CBST	1	A JOBSTATUS or DSSTATUS request was issued for all COBI jobs associated with a user, and one of the following conditions was detected: • A job number extracted from the user's catalog is found to belong to another user. • The COBI index record for a job number found in the user's catalog indicates that the record is unused. • The bit string for the COBI index indicates that a job number extracted from the user's catalog has no index record. Processing continues with the next parameter.

Communications Module	Console Code	User-Related Error Codes Explanation
M#CCDI	1	Halt I/O error on disable
	2	Abort bit during enable. This line cannot be successfully disabled and must be busied out.
M#DISK	1	Invalid key for object program
	2	Invalid parameters in call to M#DISK; treated as disk error (L#DKST=2)
M#EDIT	1	An invalid edit command code has been passed from M#ESCN.
	2	A line number or REPLACE count exceeds five digits.
	3	During editing, a program line which has no line number has been detected.
	4	A zero length count has been detected for a line in the porgram being edited.
M#ESCN	1	No match found in edit command table for command identifier found in pots
	2	Last character in edit command line was not a carrier return.
	3	End of command line pot chain reached (zero link detected) before command character count exhausted.
M#ISCAN	1	Dequeue error.
	2	Data set not opened, enqueue error.
	3	Data set not on indicated volume; may indicate that a scratch VTOC has been performed on a volume containing COBI data sets.
	4	An error has occurred during the conversion from TTR to MBBCCHHR.
	5	The last block of a fixed-block (FB) data set cannot be read.
	6	An I/O error has prevented the variable-record form line locate table from being written into the work/swap data set.
	7	An I/O error has occurred while reading a directory for a partitioned data set.
	8	An error has been detected in the OS/360 STAE exit routine; the next time this routine is entered, the CALL-OS subtask C#IOREQ will terminate abnormally.

Communications	Console	User-Related Error Codes
Module	Code	Explanation
M#ISRD	1	Buffers are assigned but there is no buffer address in the UTT.
	2	The number of buffers assigned is not equal to the buffer count.
	3	The number of input buffers assigned to this line is greater than two.
M#LDRD	2	Key field of saved program record read does not contain PROG in first four bytes.
	3	Key field of saved program record contains a different program name from that requested.
	4	Key field of saved program record read contains a different user number from that of the requester and is not a library request.
	5	Illogical situation. Trying to read the user's saved program in excess of four records.
	6	The last byte of the user's saved program read is not an EOF character (X'01').
M#LOG	2	Unable to find the user's equivalency class entry during the log-off process.
	3	Number of users indication is not positive.
M#MALC	1	Job size is larger than new job area.
	2	Job is requesting a compiler that was not initialized by the CALL/360-OS startup procedure.
M#OBJR	1	Disk error reading file descriptor record.
	2	Invalid key for file descriptor record.
	3	DCB higher than the highest in the group.
	4	DCB higher than the highest in the system.
	5	TTR conversion error.
M#OSWP	1	Unable to find the compiler to be loaded in the language table.
M#OVLY	1	A module not currently in the overlay buffer has attempted to release that buffer.
M#PASS	1	A discrepancy exists between the user number and the contents of the UGT.
	2	User number specified did not match any valid equivalency group in the system.
M#RDSO	1	More than two add-on, 24-byte core buffers attached to the UTT
	2	Program error detected while reading the user's work area from disk

Communications	Console	User-Related Error Codes
Module	Code	Explanation
M#SCR	1	Data set not dequeued. The user has requested that a data set be scratched. The data set has been closed and scratched, but an error was encountered while dequeing the data set.
	2	The volume identification in the COBI index record cannot be found in the volume identification table; actual volume identification cannot be made and the user's scratch request is denied.
	3	An I/O error has occurred while attempting to write either an updated COBI index record back on the COBI index data set or an updated JCL record back on the JCL data set.
	4	An I/O error has occurred while attempting to either read a catalog record or write an updated catalog record back on the user group data set.
M#SORT	1	The last character of the old source is not an EOF character.
	2	The last character of the current line is not a new line character and the old source is being sorted.
	3	The last character of the current line is not a new line character and the old source is <u>not</u> being sorted.
	4	The sort input and output areas overlap.
M#SSVC	1	Next compiler code not found in the language table. Language table or UTT is destroyed. Error is irrecoverable.
M#SUB	1	The last character of the current line is not a new-line character
M#TTIMR	1	A zero queue link was encountered. The link should have contained a pointer to the next UTT or a full word containing hexadecimal 'FFFFFFFF'.
	2	Accounting data was not processed for one time interval.
M#WRSO	1	A program error detected when trying to write the user's current sorted source program to disk.
	2	M#WRSO was entered by a process that was not initiated by RUN, LIST, LOAD, SAVE or any edit command.
S#INDXQ	1	Exceeded table size, too many concurrent users.
	2	A request was received to dequeue a COBI index record that had not been enqueued.

Communications Module	Console Code	User-Related Error Codes Explanation
	3	An exclusive dequeue request was issued for a UTT different from the original enqueuing UTT.
	4	A shared dequeue request was issued for an index record that was exclusively enqueued.
S#OUTR	1	An attempt was made to release the 256-byte buffer when it was not assigned.
S#RERDP	1	Either the link field of the first buffer assigned to the UTT was 0, or there was no beginning-of-line buffer address.
S#SRTR	1	An attempt was made to release the sort buffer when it was not assigned.
	2	The sort buffer was released by a UTT other than the one to which it was assigned.
	3	In the process of manipulating the sort buffer queue, a link was found that was zero.
S#SYSR	1	An attempt was made to release the system buffer when it was not assigned.
	2	The system buffer was released by a UTT other than the one to which it was assigned.
	3	In the process of manipulating the system buffer queue, a link was found that was zero.

DISK I/O ERROR MESSAGES

The form of the messages for disk errors to the communications console is as follows:

```
DISK,aaannn,bbbbbbbb,dddddd,eeeeee,ffff,text
```

where

aaannn is the six-character user number

bbbbbbbb is the eight-character program name

dddddd is the hexadecimal address of the I/O block (IOB)

eeeeee is the volume serial number

ffff is the hexadecimal cylinder and track location

text is the message text as explained below

These messages are initiated by the I#DINT module. All messages are issued after the standard OS/360 corrective routines have been tried, indicating that further error analysis is not possible. See IBM System/360 Component Descriptions - 2314 Direct Access Storage Facility and 2844 Auxiliary Storage Control (GA26-3599) for details.

Communications Console Disk I/O Error Messages (Text Only) and Explanations

COMMAND REJECT: INVALID CCW'S

Self-explanatory

CORE PROTECTION VIOLATION

Read of disk record attempted into core with different protect key than DEB.

EQUIPMENT FAILURE DURING SEEK

Self-explanatory

EQUIPMENT OR RECORDING ERROR

This message indicates that an error probably exists which affects the mechanical (rather than the electronic) parts of the equipment.

EXTENT VIOLATION

Should never happen; probably an invalid UTT if it occurs.

FILE NOT ONLINE

Device appears to be offline; probably a plug pulled or failing.

FILE PROTECT OR PROGRAM CHECK

Invalid sequence of channel command words (CCW) or file mask set to disallow WRITE or SEEK.

FLAGGED DEFECTIVE; NO RECOVERY

Flagged track in swap area or compiler area.

Communications Console Disk I/O Error Messages (Text Only) and Explanations

HOME AD. OR RECORD ZERO ERROR

Track is unusable.

RECORD LENGTH; UNFOUND; OVERFLOW

CCWs are of wrong length or record does not exist or CCW count is too long for track.

COMPILER-RELATED ERROR MESSAGES

When a program interrupt occurs, the program check PSW is decoded to determine the cause of the interrupt; this information is transmitted to the CALL-OS communications console. All such messages should be reported to the responsible systems programmer or analyst. The format of the output message is as follows:

```
USER aaannn, PROG pppppppp, LANG lll, REAS xxxxxxxx (c:hhhhh)
```

The user number, program name, language processor (compiler) name, and reason are inserted into their respective fields. Following the reason, the address portion of the PSW is transformed into the positive hexadecimal displacement (hhhhh) in either the compiler area (c has a value of L) or user area (c has a value of U) where the error occurred. A negative displacement created by a branch below the compiler or user area is indicated by an hhhhh value of zero. The reasons can be any of the following invalid program check interruptions from the compiler or user program:

```
OPERATION
PRIV. OPER.
EXECUTE
PROTECTION
ADDRESSING
SPECIFICATION
DATA
```

Other reasons that may be inserted in the reason field are listed.

Communications Console Compiler-Related Message (Reasons Only) and Explanations	Initiating Module
ERROR #151 Source pointer is destroyed while in DIM statement processor. Compilation terminated.	VERBS03
ERROR #152 Source line without line number at beginning	B$PHASE
ERROR #153 Source line without end-of-line character	B$PHASE
ERROR #154 Source line with invalid character	B$PHASE
ERROR #155 Source pointer is destroyed while in an identifier scan routine.	NUCLEUS
ERROR #170 Temporary storage counter below limit	FORMULA
ERROR #171 Unknown operator (binary)	FORMULA

Communications Console Compiler-Related Message (Reasons Only) and Explanations	Initiating Module
ERROR #172 Unknown operator (unary)	FORMULA
ERROR #173 Undefined identifier type	FORMULA
ERROR #174 Undefined delimiter	FORMULA
ERROR #175 Stack underflow	FORMULA
ERROR #331 A source statement was encountered that was not preceded by a line number. User job terminated.	FORTRAN
EXCEEDED CORE More core was requested than was available for the current run of CALL-OS.	M#SSVC
ILLEGAL SVC Compiler has issued an SVC with an undefined or unimplemented code type.	M#SSVC
P CHK IN TRAP A program check has occurred in an interrupt routine. A message describing the program check has already been printed.	M#SSVC
PSLTH TOO BIG Number of relocatable addresses to be updated is unreasonable (extends beyond user's upper memory boundary).	M#DISP
PSPTR<X'550' Block of relocatable addresses is out of bounds (erroneously points to communications region).	M#DISP
PSW1SV ERROR User's current PSW points outside the user's allocated memory boundaries.	M#DISP
SVC 2 ERROR SVC Code 2 is issued and an incorrect BUFPTR value is detected.	M#SSVC

Communications Console Compiler-Related Message (Reasons Only) and Explanations — Initiating Module

SVC 3/4 ERROR — M#SSVC

User's communications region is unacceptable. In particular, FILENBR is outside the range 1-4, FILEPTR has a displacement value beyond this user's program area, or an attempt was made to read or write past the end of the file.

SVC 5 ERROR — M#SSVC

User is returning too much core.

SVC 7/8 ERROR — M#SSVC

The user has requested an exit from the arithmetic trap routine to an address beyond his limits. (PSW2SV is out of bounds.)

SVC 11 ERROR — M#SSVC

Compiler generated a program that issued an SVC 11.

SVC 17 ERROR — M#SSVC

An SVC 17 has occurred when phase 1 of a compilation is not in process.

SVC 21 ERROR — M#SSVC

User's communications region is unsuitable for performance of the open function. Any one of the following may be true:

- Open request was received for a file previously opened but never closed.
- Open bit set but neither as an input file nor as an output file.
- Data file table extends beyond the user's program area.
- No file found with open bit set.
- The file table pointer table has no end-of-table marker.

During execution of the CALL-OS utilities, messages are printed at the OS/360 system printer. This printer is the device defined by either a SYSPRINT or PRINTA DD statement when the utility is run. The messages are in alphabetical order either according to an identifier of the form DIBxxxxx or, if no identifier is present, according to the first word of message text.

An identifier indicates the utility program which issued the message, and in some cases, which module within the utility. The following identifiers are used:

DIBDBynn — Indicates that the message was issued by the data base utility, DIBCADBU. In these messages, the ynn is the message number and y identifies the module which issued tne message as follows:

Value of y	Module(s) and Corresponding Function
0	DIBINIT, DIBINTER, DIBOPEN, DIBTERM, DIBSUB, DIBDISK, DIBGDSK, DIBMESG, DIBTTOT
1	DIBREORG - REORGANIZE function
2	DIBVALDT - VALIDATE function
3	DIBDELET - DELETE function
4	DIBWRITE - WRITE function
5	DIBTAPE - TAPE function
6	DIBINSRE - INSERT/REPLACE function Note: When INSERT/REPLACE is used with OPTIONS=VAL, the VALIDATE function is used before INSERT/REPLACE; therefore, DIBDB2nn messages may precede DIBDB6nn messages in the output.
7	DIBRECON - RECONSTRUCT function
8	DIBACCNT - ACCOUNT function
9	DIBJOBFD - JOBFIND function

DIBUT2nn — Indicates that the message was issued by U#UTIL2 during the system build process

DIBUT5nn — Indicates that the message was issued by U#UTIL5 while processing COBI data sets

DIBUT6nn — Indicates that the message was issued by DIBCONPR, the COBI utility which converts cataloged procedures to a form that can be used to scan COBI output data sets

*** COMPILER - BAD ATTRIBUTE SWITCH ON LOAD MODULE ***

This message is issued by U#UTIL1; it indicates an incorrect attribute switch in the partitioned data set directory entry for the load module version of the compiler specified by the dd name of the DD card being processed. The second byte of the attribute switch should have ones in bits one and two (the linkage editor assigned origin of first block of text is zero and entry point assigned by linkage editor is zero).

*** COMPILER - BAD BLDL ON COMPILER NAME ***

This message is issued by U#UTIL1; it indicates that the execution of the Build-a-List macro instruction was unsuccessful when referencing the compiler indicated by the dd name on the DD card being processed. In this case, there is probably no compiler of that name in the partitioned data set specified by the LANG DD card.

*** COMPILER - BAD OPEN LOAD MODULE LIBRARY (LANG)

This message is issued by U#UTIL1; it indicates that OS/360 was unable to open (for input) the partitioned data set specified by the LANG DD card, which contains the load module versions of the compilers. This partitioned data set is needed in order to write a fast-loading version of a compiler (specified by the ddname of the DD card being processed) into the data set specified by the dsname of the DD card being processed.

*** COMPILER - BAD OPEN OUTPUT DATA SET ***

This message is issued by U#UTIL1; it indicates that OS/360 was unable to open (for output) the data set specified (by the DD card being processed) in order to write a fast-loading version of a compiler on it.

*** COMPILER - INVALID RECORD SIZE ***

This message is issued by U#UTIL1; it indicates that a control record in the load module version of the compiler being processed has an invalid record size. Relink edit the compiler and process the new load module with U#UTIL1.

*** COMPILER - INVALID RECORD TYPE ***

This message is issued by U#UTIL1; it indicates that a control record in the load module version of the compiler being processed is not a valid type, that is, it is not an RLD or CTL record. Relink edit the load module and process the load module with U#UTIL1.

*** COMPILER - PERMANENT I/O ERROR ***

This message is issued by U#UTIL1; it indicates that a permanent I/O error has occurred during either the reading of the load module version of a compiler or the writing of the fast-loading version. This is most likely a hardware problem.

*** COMPILER - COMPILER TOO BIG FOR DS ***

This message is issued by U#UTIL1; it indicates that the data set specified by the ddcard being processed is not large enough to contain the fast-loading version of the specified compiler. A larger data set must be allocated. Remember that multiple extents are not allowed for compiler data sets.

DIBDB001 ERROR - DSNAME ON DDCARD-dddddddd- DOES NOT MATCH DSNAME IN INDEX.

The data set name specified on the indicated ddcard in the job control language is not the same as the data set name specified for that ddcard in the CALL-OS Index. Correct JCL and rerun the job. Condition code = 4.

DIBDB002 ERROR - DUPLICATE DDCARDS FOUND IN JOB CONTROL LANGUAGE.

More than one ddcard with the same ddname was found in the job control language. Correct JCL; rerun the job. Condition code = 4.

DIBDB003 GROUP aaabbb NOT INCLUDED IN REORG; NO DDCARD FOUND IN JCL.

Not all groups in the "from" cluster which overlap the "to" group are included in the reorganization because no ddcard was supplied for those groups. Condition code = 4.

DIBDB004 ERROR - DATA SETS REQUIRED FOR PROCESSING ARE RESERVED TO ANOTHER JOB.

An attempt to enqueue the user and/or system group data sets required for the utility control statement indicated that the data sets have been enqueued either by CALL-OS or another data base utility job. Those control statements not processed should be rerun at a later time when no other job is using the data sets. The control statement on which the error occurred is not processed, and processing continues with the next control statement. Condition code = 8.

DIBDB005 ERROR - DATA SETS REQUIRED FOR REORG OR RECON NOT INITIALIZED.

The output data sets for the REORGANIZE and RECONSTRUCT functions must be newly created, initialized data sets. See the _CALL-OS Executive and Utilities Program Description Manual_ for a description of the use of the UTILX and U#UTIL1 utilities to preformat the data sets. The control statement on which the error occurred is not processed, and processing continues with the next control statement. Condition code = 8.

DIBDB006 ERROR - DUPLICATE PARAMETERS ON CONTROL STATEMENT.

The same parameter appears more than once on the control statement being processed. Correct control statement and rerun. The control statement on which the error occurred is not processed, and processing continues with the next control statement. Condition code = 8.

DIBDB007 UNABLE TO CONTINUE PROCESSING THIS CONTROL STATEMENT.

An error which prevented normal processing occurred in either the VALIDATE function or the TAPE function when being executed in conjunction with the INSERT/REPLACE function or DELETE function, respectively. The error caused the INSERT/REPLACE function or DELETE function not to be processed. Correct the problem causing the original error and rerun. The control statement on which the error occurred is not processed, and processing continues with the next control statement. Condition code = 8.

DIBDB008 ERROR - CARD SHOULD HAVE ./ IN COLUMNS 1 AND 2.

A control statement card was expected containing ./ in the first two columns. Correct control statement card and rerun. The control statement on which the error occurred is not processed, and processing continues with the next control statement. Condition code = 8.

DIBDB009 ERROR - INVALID PARAMETER ON ABOVE xxxxxxxxx.

When xxxxxxxxx is CARD, the message indicates a syntactical error on the card printed immediately above the message. This error could be an invalid keyword parameter, parameter value, or delimiter, or the parameters exceed column 71. Correct error on card and rerun. When xxxxxxxxx is STATEMENT, the message indicates a parameter inconsistency on the control statement associated with the change. Check the valid parameter combinations for the function requested and correct the control statement. In either case, the control statement on which the error occurred is not processed, and processing continues with the next control statement; condition code = 8.

DIBDB010 NO PARAMETERS ON CONTROL STATEMENT.

A ./ card has been read that contains no parameters. The card is ignored. Condition code = 4.

DIBDB011 ERROR - INVALID OR NO FUNCTION SPECIFIED.

The first field following the ./ of the control statement does not contain a valid data base utility function name. Correct the control statement and rerun. The control statement on which the error occurred is not processed, and processing continues with the next control statement. Condition code = 8.

DIBDB012 ERROR - INVALID OR NO xxxxxxxx PARAMETER SPECIFIED.

For a function requiring the named parameter, either the parameter was not given on the control statement or the parameter was given but did not specify a valid parameter value. Correct the control statement and rerun. The control statement on which the error occurred is not processed, and processing continues with the next control statement. Condition code = 8.

DIBDB013 ERROR - REQUIRED DATA SETS NOT INCLUDED IN JCL.

No ddcard was found in the job control language for those data sets required by the control statement. Include ddcards in JCL and rerun. The control statement on which the error occurred is not processed, and processing continues with the next control statement. Condition code = 8.

DIBDB014 ERROR - NO INDEX ENTRY FOUND FOR TO GROUP.

The group indicated by either the USER parameter or the USRGROUP parameter is not specified by any of the CALL-OS Index entries. Ensure that the USER parameter specifies a user number within a CALL-OS user or system group or that the USRGROUP parameter exactly names a CALL-OS user or system group. The control statement on which the error occurred is not processed, and processing continues with the next control statement. Condition code = 8.

DIBDB015 ERROR - NO INDEX ENTRY FOUND FOR FROM GROUP.

The group indicated by either the FROMUSER parameter or the FRMGROUP parameter is not specified by any of the CALL-OS Index entries. Ensure that the FROMUSER parameter specifies a user number within a CALL-OS user or system group or that the FRMGROUP parameter exactly names a CALL-OS user or system group. The control statement on which the error occurred is not processed, and processing continues with the next control statement. Condition code = 8.

DIBDB016 HIGHEST CONDITION CODE WAS nn.

The listed code is the highest generated in the job step.
00 - no errors encountered.
04 - an error was encountered that did not terminate processing of a control statement.
08 - an error was encountered that did terminate processing of a control statement.

DIBDB017 END OF CALL-OS DATA BASE UTILITY RUN.

The data base utility program has completed processing.

DIBDB030 ERROR - UNABLE TO OPEN DATA SET; DDNAME=dddddddd.

Unable to open the data set specified by the named ddcard. Condition code = 8.

DIBDB031 ERROR - BAD DEB; DDNAME=dddddddd.

After a successful open, the DCB pointer to the data extent block data extent block (DEB) points to an area that does not have hexadecimal 'F' in the low-order four bytes of the DEB identifier (DEB+24). Condition code = 8.

DIBDB032 ERROR RETURN FROM OS CONVERT ROUTINE; DDNAME=dddddddd.

Use of the OS/360 routine to convert a relative track address to absolute (pointed to by the Communication Vector Tables plus 28) resulted in an error return code. Routine was unable to convert the relative disk address to an absolute address. Condition code = 8.

DIBDB033 USER GROUP DATA SET DEPLETED; DDNAME=dddddddd.

There is no more disk space available for allocation in this data set. Condition code = 4.

DIBDB034 ERROR - INDEX ENTRIES OUT OF SEQUENCE; GROUP=aaabbb.

Resequence CALL-OS Index entries according to their type (user group, system group, compiler, etc.), then according to group names, and finally according to the relative data set number within the group. Condition code = 8.

DIBDB035 NO GROUP BEING OPENED.

No group was opened during this pass through DIBOPEN. This error should not occur since earlier checks in DIBINIT should discover the error. Condition code = 4.

DIBDB036 ERROR - BOTH SYSGRP'S TO BE OPENED.

During one pass through DIBOPEN, both system groups should not be opened since the first pass opens only "to" group data sets and the second pass opens only "from" cluster data sets. This should not occur and indicates an error in DIBINIT. Condition code = 8.

DIBDB037 PHYSICAL I/O ERROR READING xxxxxxxxxxxx; DDNAME=dddddddd.

A read request for the specified record type using DIBDISK returned with an indicated disk read error. This could be a wrong length record or invalid read parameter indication. Condition code = 8.

DIBDB038 K or D LENGTH ERROR READING xxxxxxxxxxxx; DDNAME=dddddddd.

This indicates a KEY or DATA length error was encountered by DIBDISK in attempting to read the specified record type. Condition code = 8.

DIBDB039 ERROR - INVALID CODE IN DIBINTER.

DIBOPEN was entered from DIBINIT and found a code other than '01' or '02' in G#FLAG1 of DIBINTER. This situation should not occur and indicates an error in DIBINIT or DIBOPEN. Condition code = 8.

DIBDB040 ERROR - DATA SET OTHER THAN SYSTEM GROUP OR USER GROUP TO BE OPENED.

A CALL-OS Index entry of type other than system group or user group was flagged for opening. This situation should not occur and indicates an error in DIBINIT or DIBOPEN. Condition code = 8.

DIBDB041 ERROR - NOT ALL DATA SETS FOR GROUP aaabbb ARE AVAILABLE.

Not all of the data sets for the named group are available for processing. Ensure that DD statements for all data sets for the group are included in the JCL. Condition code = 8.

DIBDB045 ERROR WRITING ALLOCATION RECORD; GROUP=aaabbb, DSNUMBER=nn.

An error was encountered by DIBDISK in attempting to write the allocation record for a data set in the specified group. The relative data set number of the data set is specified. Condition code = 4.

DIBDB081 INTERNAL DATA FILE FORMAT VIOLATION; REMAINDER OF RECORD CANNOT BE PRINTED.

The data file in INTERNAL format currently being printed in formatted form contains incorrect or inconsistent data type codes or replication factors. The erroneous data file may be inspected in detail by using the data base utility WRITE function with the HEX option. Condition code = 4.

DIBDB111 ERROR - PARTITION/REGION NOT LARGE ENOUGH FOR REORG INTO GROUP aaazzz, CLUSTER k; FUNCTION TERMINATED.

The requested reorganization into the indicated group and cluster could not be completed due to lack of main storage. The function should be resubmitted with a larger partition or region specified for the data base utility. Condition code = 8.

DIBDB151 I/O ERROR ON INPUT GROUP aaazzz AT dd ttt rr (DTTR). RECORD TYPE SHOULD BE cccc[, LENGTH pppp]

An error was detected while trying to read a record from the indicated location within the input group. This message is followed by a hexadecimal dump or formatted listing of the record in error. This information is in turn followed by one or more messages which indicate any further action taken by the REORGANIZE function. Condition code = 4.

DIBDB152 KEY FORMAT ERROR. RECORD TYPE SHOULD BE cccc, USER GROUP WAS aaazzz, DTTR WAS dd tttt rr.

This message indicates that a record key field is not formatted correctly. The type code in the catalog entry does not match the type code in the corresponding record. All record-key, user-number fields are checked for a proper user number. This message is followed by a hexadecimal dump or formatted listing of the record in error. This information is in turn followed by one or more messages which indicate any further action taken by the REORGANIZE function. Condition code = 4.

DIBDB155 ERROR - USER NUMBERS IN aaam00 DIRECTORY RECORD NOT IN EQUIVALENCY FILE.

One or more user numbers found in the indicated directory cannot be found in the equivalency file for the group. The directory entry is dropped. Condition code = 4.

DIBDB159 I/O ERROR ON OUTPUT GROUP aaazzz AT dd tttt rr (DTTR).

An error was detected while trying to write a record at the indicated location within the output group. This message is followed by a hexadecimal dump or formatted listing of the record in error. This information is followed by the termination message DIBDB191. Condition code = 8.

DIBDB161 PRECEDING RECORD (FILE UNIT iii OF DATA FILE xxxxxxxx FOR USER aaammm) WAS CLEARED AND WRITTEN TO USRGROUP.

This message follows a hexadecimal dump or formatted listing of the indicated record. This record either caused an input error or contained an invalid key. The record in error is cleared to X'01' characters, given a valid key and empty data record control bytes, and written into the reorganized data set. The original contents of the record are lost. Processing continues with the next data record in the file. Condition code = 4.

DIBDB163 THE ABOVE CATALOG ENTRY FOR USER aaammm HAS BEEN DROPPED DUE TO INPUT ERROR OR INVALID KEY.

This message follows a hexadecimal dump or formatted listing of a catalog entry for the indicated user number. The catalog entry points to a record which either could not be read or failed the record key validity check. The catalog entry is nullified and the record in error is not moved to the reorganized data set. Condition code = 4.

DIBDB165 ALL PROGRAM AND DATA FILES FOR USER aaammm HAVE BEEN DROPPED DUE TO INPUT ERROR OR INVALID KEY.

The first link of the catalog for the indicated user number could not be read. None of the program and data files pointed to by the catalog can be moved to the reorganized data set. Condition code = 4.

DIBDB166 ALL FILES AFTER xxxxxxxx FOR USER aaammm HAVE BEEN DROPPED DUE TO INPUT ERROR OR INVALID KEY.

At least one link of the catalog for the indicated user number has been read, but an error occurred while reading another link. The catalog is truncated after the indicated file name; program or data files pointed to by subsequent catalog entries are not moved to the reorganized data set. Condition code = 4.

DIBDB171 SYSLIB EQUIVALENCY ENTRY IN OUTPUT GROUP HAS BEEN GIVEN DEFAULT PASSWORD DUE TO INPUT ERROR OR INVALID KEY.

The equivalency file for the system group could not be read. The system group is assigned a default password of SECURITY. Condition code = 4.

DIBDB175 ALL EQUIVALENCY ENTRIES FROM INPUT GROUP aaazzz HAVE BEEN DROPPED DUE TO INPUT ERROR OR INVALID KEY.

The first link of the equivalency file for the indicated user group could not be read. None of the user numbers in the group will be present in the reorganized data base and information associated with them is not moved to the reorganized data base. Condition code = 4.

DIBDB176 ALL EQUIV ENTRIES FROM GROUP aaazzz AFTER USER aaammm HAVE BEEN DROPPED DUE TO INPUT ERROR OR INVALID KEY.

At least one link of the equivalency file for the indicated user group has been read, but an error occurred while reading another link. The file is truncated after the indicated user number; subsequent user numbers and the associated information will not be present in the reorganized data base. Condition code = 4.

DIBDB181 ERROR - OUT OF SPACE ON GROUP aaazzz, USER aaammm; REORG TERMINATED.

The data set for the indicated user group is full; the information associated with the indicated user number cannot be written. The reorganization process is terminated. Condition code = 8.

DIBDB191 REORG TERMINATED ABNORMALLY DUE TO NON-RECOVERABLE I/O ERROR.

An I/O error has occurred and normal recovery measures have failed. The reorganization process is terminated. Condition code = 8.

DIBDB201 ERROR - THE AAN OF USER2 MUST BE SAME AS THAT OF USER.

The user numbers for USER and USR2 do not fall in the same sub group. Condition code = 8.

DIBDB202 ERROR - THE USER MUST BE LESS THAN OR EQUAL TO USR2.

The specification for USR2 was lower than that for USER; however, the USER specification must represent the low end of the range. Condition code = 8.

DIBDB203 ERROR - USR2 IS NOT OF PROPER ALPHANUMERIC FORM.

The sixth character of the user number is nonnumeric. Condition code = 8.

DIBDB204 ERROR - INVALID OR NO xxxxxxxx.

Either the PASSWORD (or USERPASS) parameter was not specified, or the parameter was specified but contained a leading blank, or the password specified did not match the password in the SYSLIB equivalency record. Condition code = 8.

DIBDB205 ERROR - SYSLIB EQUIVALENCY ENTRY COULD NOT BE FOUND.

An equivalency record for the SYSLIB user could not be found in the SYSGRP data set. This is a system error which must be corrected before the VALIDATE function can be used. Condition code = 8.

DIBDB206 ERROR - I/O ERROR IN xxxxx OF yyyyyyyy EQUIVALENCY FILE.

An I/O error occurred either during a read of an equivalency record for SYSLIB or FROMUSER, or during a read or write of an equivalency record for a USER. Partial execution of the VALIDATE function may have been completed. Resubmit the control statement. Condition code = 8.

DIBDB207 ERROR - UGT FOR xxxxxxxx COULD NOT BE FOUND.

An entry in the user group data set translation table could not be found for the specified USER, FROMUSER, or SYSLIB. Either no DD statement was supplied for the user group or system group, or the wrong DD statement was supplied for the system group. Condition code = 8.

DIBDB208 ERROR - USER MAY NOT = SYSLIB OR **LIB WHEN OPTIONS=VAL.

The USER parameter specified either SYSLIB or **LIB; however, SYSLIB is always validated and **LIB may not be validated. Condition code = 8.

DIBDB209 ERROR - FROMUSER MUST BE SPECIFIED WHEN INPUT= DISK OR TAPE.

Validation was requested for the INSERT/REPLACE function when INPUT=DISK or TAPE; the FROMUSER must be specified to permit equivalent from user numbers to be validated for the to user range. Condition code = 8.

DIBDB210 ERROR - FOR DISK OR TAPE INPUT, WHEN USER=AAAM00, FROMUSER MUST = BBBR00.

User aaam00 represents a * directory into which entries are to be made; the entries are taken from the FROMUSER * directory. Condition code = 8.

DIBDB211 ERROR - FOR A USER RANGE, USER AND FROMUSER MUST HAVE THE SAME LAST TWO DIGITS.

To check for all equivalent FROMUSER user numbers, FROMUSER must start with the same user number as USER. Condition code = 8.

DIBDB212 ERROR - USER MUST BE IN SUB GROUP SERVED BY FROMUSER DIRECTORY.

For a single user to access directory information, his user number must be within the range of user numbers served by the sub group directory. Condition code = 8.

DIBDB213 ERROR - FOR CARD OR OSDS INPUT, USER MAY NOT = AAAN00.

The USER parameter specified a user number of the form aaan00. Programs or data files specified as CARD or OSDS input may not be entered into a * directory. Condition code = 8.

DIBDB214 ERROR - FOR CARD OR OSDS INPUT, NO USR2 MAY BE SPECIFIED.

For CARD or OSDS input, only one user can be processed at a time. Condition code = 8.

DIBDB215 ERROR - NO EQUIVALENT FROMUSER(S) FOUND.

For the specified USER (or USER range), no matching user number(s) was found validated in the FROMUSER equivalency file. Condition code = 8.

DIBDB216 ERROR - USER aaannn ALREADY VALIDATED.

The specified user number was already validated. For a range of users, condition code = 0; for a single user, condition code = 8.

DIBDB217 ERROR - NO DISK SPACE AVAILABLE.

Disk storage was not available for a newly-generated catalog or equivalency record. Condition code = 8.

DIBDB218 ERROR - WRITE FAIL OF USER aaannn CATALOG.

An I/O error occurred during an attempt to write a catalog record for the specified user. Condition code = 8.

DIBDB220 USER aaannn VALIDATED.

The specified user number was successfully validated. Condition code = 0.

DIBDB221 ERROR - TAPEIN OPEN FAILED.

The TAPEIN data set containing the tape FROMUSER group could not be opened. The DD card may be missing. Condition code = 8.

DIBDB225 ERROR - GETMAIN FOR nnnn BYTES FAILED.

Main storage in the amount specified was not available for buffer use. Condition code = 8.

DIBDB301 ERROR - FROMUSER=xxxxxx MAY NOT BE GIVEN WHEN CANCEL OR PULL IS SPECIFIED.

In this message, xxxxxx is either SYSLIB or **LIB. For CANCEL, both are invalid; the SYSLIB user may not be cancelled and FROMUSER=**LIB has no meaning for CANCEL. For PULL, only SYSLIB is invalid because FROMUSER=**LIB is specified to pull entries from the **Directory regardless of user number. Condition code = 8.

DIBDB302 ERROR - THE AAAN OF FROMUSR2 MUST BE THE SAME AS THAT OF FROMUSER.

The range of users is restricted to a subscription group. Condition code = 8.

DIBDB303 ERROR - THE FROMUSER MUST BE LESS THAN OR EQUAL TO THE FROMUSR2.

FROMUSER is the lower end and FROMUSR2 the upper end of the range of users. Condition code = 8.

DIBDB304 ERROR - FROMUSR2 IS NOT OF PROPER ALPHANUMERIC FORM.

The sixth character of the user number is nonnumeric. Condition code = 8.

DIBDB305 ERROR - SYSGRP COULD NOT BE OPENED TO CHECK SYSLIB PASSWORD.

No SYSGRPnn DD statement was included in the JCL so the SYSLIB user files within the data set could not be accessed. Condition code = 8.

DIBDB306 ERROR - SYSLIB EQUIVALENCY ENTRY COULD NOT BE FOUND.

No SYSLIB user equivalency record was found within the SYSGRP00 data set. This is a system error which must be corrected before the DELETE function can be used. Condition code = 8.

DIBDB307 ERROR - NO COMMAND WAS INPUT.

No command choice was made so no specific delete action could be taken. Condition code = 8.

DIBDB308 ERROR - THE INPUT xxxxxx PASSWORD AND xxxxxx EQUIVALENCY PASSWORD DID NOT MATCH.

The appropriate password, where xxxxxx is either SYSLIB or USER was not input, so no action could be taken. The SYSLIB password is needed for CANCEL and for PULL and PURGE of a range of users or a single cancelled user. The USER password is needed for PURGEs and PULLs of a single user not cancelled. Condition code = 8.

DIBDB309 ERROR - THE UGT FOR THE FROMUSER COULD NOT BE FOUND.

Either the user group containing the FROMUSER was not opened because its ddcard was not included in the JCL or the user group is not in the data base. Condition code = 8.

DIBDB310 ERROR - AN I/O ERROR OCCURRED DURING A xxxxx OF THE FROMUSER EQUIVALENCY FILE.

In this message, xxxxx is either WRITE or READ. Partial execution of the command may have taken place prior to the write or read error. Resubmit control statement. Condition code = 8.

DIBDB311 ERROR - USER NUMBER aaannn COULD NOT BE FOUND IN THE EQUIVALENCY FILE.

The user whose number is aaannn was not a validated user. Condition code = 8.

DIBDB312 ERROR - NO USER NUMBERS IN THE RANGE aaannn TO aaannn WERE FOUND IN THE EQUIVALENCY FILE.

User range input contained no validated users. Condition code = 8.

DIBDB313 ALL USER NUMBERS IN THE RANGE aaannn TO aaannn WERE CANCELLED.

The input range of users was successfully cancelled. Condition code = 0.

DIBDB314 USER NUMBER aaannn HAS BEEN CANCELLED.

The input user was successfully cancelled. Condition code = 0.

DIBDB315 ERROR - PURGE AND PULL CANNOT BOTH BE SPECIFIED.

Both PURGE and PULL were specified on the control statement; however, when a file is purged, the name is automatically pulled from the directory. Condition code = 8.

DIBDB316 ERROR - FOR PURGE OR PULL FOR A SINGLE USER, NAME MUST BE SPECIFIED.

A filename, a prefix, or (ALL) must be specified for the purging or pulling of a file for a single user. Condition code = 8.

DIBDB321 ERROR - I/O ERROR IN xxxxx OF USER aaannn CATALOG.

In this message xxxxx is either READ or WRITE. Partial execution of the command may have taken place prior to the read or write error. Resubmit function card. Condition code = 8.

DIBDB322 ERROR - USER aaannn FILE nnnnnnnn NOT FOUND OR INVALID TYPE OR LOCKED.

Either the input file of the type specified could not be found in the input user's catalog or the file was locked and OPTIONS=UNLOCK was not specified. Condition code = 8.

DIBDB323 ERROR - NO FILES OF THE TYPE SPECIFIED WERE FOUND FOR USER aaannn.

The catalog of the input user contained no files of the type specified. Condition code = 8.

DIBDB324 ALL FILES OF THE TYPE SPECIFIED WERE PURGED FOR USER aaannn.

The input user files were successfully purged. Condition code = 0.

DIBDB325 EVERY USER FILE IN THE USER RANGE aaannn TO aaannn WAS PURGED.

The purge was successful. Condition code = 0.

DIBDB326 USER aaannn FILE xxxxxxxx yyyyyy.

Either the user file of the type specified was successfully purged or the file was locked and OPTION=UNLOCK was not specified. Condition code = 0.

DIBDB330 ERROR - FOR PULL, IF FROMUSER=AAAN00, FROMUSR2 MUST NOT BE INPUT.

If PULL is specified, a FROMUSER of the form aaan00 specifies the directory from which the files are to be pulled, not the first user number of a range of user numbers. Condition code = 8.

DIBDB331 ERROR - FAIL IN xxxxx OF yy DIRECTORY.

An I/O error occurred during a read or write of either the * or ** directory. Condition code = 8.

DIBDB332 ERROR - NO FILES WERE PULLED FROM THE yy DIRECTORY.

No file(s) of the type and for the user specified in the control statement were found in the * or ** directory. Condition code = 8.

DIBDB333 ALL yy DIRECTORY FILES WERE PULLED FOR USERS IN THE RANGE aaannn TO aaannn.

All files for the specified range of users were pulled from the * or ** directory. Condition code = 0.

DIBDB334 ALL yy DIRECTORY FILES OF THE TYPE SPECIFIED WERE PULLED FOR USER aaannn.

The specified user's files were pulled from the * or ** directory. Condition code = 0.

DIBDB335 USER aaannn FILE xxxxxxxx PULLED FROM yy DIRECTORY.

The file specified was pulled from the * or ** directory. Condition code = 0.

DIBDB340 ERROR - GETMAIN FOR nnnn BYTES FAILED.

Enough core was not available for program processing use, where nnnn is the number of bytes. Condition code = 8.

DIBDB341 ERROR - INVALID NAME.

The NAME value specified contained a leading blank. Condition code = 8.

DIBDB351 WARNING - THE xxxx PARAMETER SHOULD NOT BE SPECIFIED WHEN CANCEL IS THE ONLY COMMAND GIVEN - IGNORED.

In this message, xxxx is FILENAME or DATE. These parameters have no meaning for the CANCEL command. Condition code = 4.

DIBDB352 ERROR - FOR PURGE, FILE MAY NOT = NULL.

FILE=NULL is for PULL only; it refers to directory entries which have no corresponding catalog entries. Condition code = 8.

DIBDB353 ERROR - OPTIONS MAY NOT = * OR ** WHEN FROMUSER = **LIB OR AAAN00.

For PULL, OPTIONS=* or ** specifies the directory from which entries are to be pulled for the users specified by FROMUSER. However, FROMUSER=aaan00 or **LIB also specifies the directories and thereby makes the options specification an error. Condition code = 8.

DIBDB354 ERROR - FOR PULL, FILE MAY NOT = JOB OR EVERY.

COBI job entries may not be pooled. Thus, FILE=JOB or EVERY (which includes JOB) may not be specified for PULL. Condition code = 8.

DIBDB355 ERROR - IF DATE IS INPUT, FILE MAY NOT = JOB OR EVERY.

FILE=JOB or EVERY (which includes JOB) refer to COBI jobs; however, these jobs do not have a last-used date in their catalog entries. Condition code = 8.

DIBDB356 ERROR - FOR PURGE, FROMUSER MAY NOT = **LIB.

**LIB refers to a directory; directory entries are pulled, not purged. Condition code = 8.

DIBDB357 WARNING - OPTION INVALID FOR COMMAND.

Certain OPTIONS parameters may be specified only with certain commands. For example, OPTIONS=NOTAPE or UNLOCK may be specified only with the PURGE command; OPTIONS=* or ** may be specified only with the PULL command. Condition code = 4.

DIBDB360 ERROR - OPEN FAIL OF COBI xxxxx DATA SET.

Either the COBI index or COBI JCL data set could not be opened. Ensure that the DD statement is present and specified correctly. Condition code = 8.

DIBDB361 ERROR - OPEN FAIL OF SYSPUNCH DATA SET.

The card punch data set could not be opened. Ensure that the DD statement is present and specified correctly. Condition code = 8.

DIBDB362 ERROR - CATALOG ENTRY AND COBI INDEX ENTRY DO NOT MATCH FOR
USER aaannn ENTRY xxxxxxxx.

The COBI index record referred to by the job number in the catalog entry was either not active, belonged to a different user, or had a different COBI job name. The JOBFIND function should be executed to update the catalog. Condition code = 4.

DIBDB401 ERROR - BAD OPEN OF dddddddd DATA SET [; WRITE TERMINATED.]

The data set defined by the specified DD statement could not be opened and the output specified by OUTPUT=CARD or OUTPUT=OSDS could not be produced; in this case, condition code = 4. If dddddddd is either SYSPUNCH or LIBRARY, the WRITE function is terminated; in this case, condition code = 8. In all cases, check the JCL to ensure that the required DD statement is present and specified correctly. Correct the JCL and resubmit the function request.

DIBDB402 ERROR - DATA SET SPECIFIED BY dddddddd HAS INVALID
CHARACTERISTICS [; WRITE TERMINATED.]

The data set defined by the specified DD statement was opened, but the DCB parameters of the data set did not satisfy the WRITE function requirements; in this case, condition code = 4. If dddddddd is LIBRARY, the WRITE function is terminated; in this case, condition code = 8. See the publication CALL-OS Executive and Utilities Program Description Manual for a discussion of the required DCB attributes. Correct the JCL and resubmit the function request.

DIBDB411 ERROR - PARTITION/REGION NOT LARGE ENOUGH FOR WRITE; FUNCTION TERMINATED.

The WRITE function could not be executed due to insufficient main storage. Specify a larger partition or region for the data base utility and resubmit the function. Condition code = 8.

DIBDB421 ERROR - REQUIRED xxxx PARAMETER NOT SPECIFIED FOR WRITE; FUNCTION TERMINATED.

The user has failed to specify a control statement parameter which is required by the WRITE function. Consult the publication CALL-OS Executive and Utilities Program Description Manual for a discussion of required parameters. Some parameters are required in all cases, while others may be required only in conjunction with certain parameters. Condition code = 8.

DIBDB431 ERROR - INCORRECT xxxxxxx PARAMETER SPECIFIED FOR WRITE; FUNCTION TERMINATED.

Either a control statement parameter has been specified with an unallowable value, or an invalid combination of parameters has been given. See the publication CALL-OS Executive and Utilities Program Description Manual for allowable parameters and parameter combinations. Condition code = 8.

DIBDB441 ERROR - FROMUSER SPECIFIED COULD NOT BE FOUND; WRITE TERMINATED.

The FROMUSER parameter on the function control statement either specified a user number in a user group which was not available in the specified cluster, or, if the user group was available, the specified user number could not be found in the equivalency file for the group. Condition code = 8.

DIBDB442 THE ** LIBRARY HAS NO EQUIVALENCY ENTRY. THE FIRST DIRECTORY LINK IS AT 00 0002 02 IN SYSGRP.

The user has specified parameters which indicate that an equivalency entry is to be printed for the **Library. This message indicates that the system is designed so that the first directory link for the **Library is always located at DTTR 00 0002 02 in the system group data set, and that there is no equivalency entry in the CALL-OS system for the **Library. Condition code = 0.

DIBDB445 ERROR - REQUIRED PASSWORD INCORRECT OR NOT SPECIFIED; WRITE TERMINATED.

The PASSWORD parameter is required on the control statement in order to perform the requested WRITE function; either the PASSWORD parameter was not specified, or the specified password did not match the corresponding password in the equivalency file, or the SYSLIB password is required and the system group in the specified cluster was not available (no SYSGRPnn DD statements were provided). Condition code = 8.

DIBDB451 I/O ERROR OR INCORRECT KEY DETECTED WHEN READING SYSTEM FILE LINK FOR FROMUSER OR FRMGROUP.

Either a permanent I/O error occurred when reading an equivalency file, catalog, or directory record for the FROMUSER or FRMGROUP specified on the control statement, or else the record was read but was found to contain an invalid key field. These conditions generally indicate that the data base has been invalidly altered; however, the cause of a permanent I/O error may be a hardware malfunction. If the problem persists, contact the installation system programmer. Condition code = 4.

DIBDB452 I/O ERROR OR INCORRECT KEY DETECTED WHEN READING SYSTEM FILE LINK FOR USER aaannn.

Either a permanent I/O error occurred when reading an equivalency file, or catalog for user aaannn (pointed to by an entry in the directory specified by the FROMUSER parameter), or else the record was read but was found to contain an invalid key field. These conditions generally indicate that the data base has been invalidly altered; however, the cause of a permanent I/O error may be a hardware malfunction. If the problem persists, contact the installation system programmer. Condition code = 4.

DIBDB453 I/O ERROR OR INCORRECT KEY DETECTED WHEN READING FILE DESCRIPTOR FOR xxxxxxxx

Either a permanent I/O error occurred when reading a file descriptor record, or else the record was read but was found to contain an invalid key field. These conditions generally indicate that the data base has been invalidly altered; however, the cause of a permanent I/O error may be a hardware malfunction. If the problem persists, contact the installation system programmer. Condition code = 4.

DIBDB454 I/O ERROR OR INCORRECT KEY DETECTED WHEN READING FILE xxxxxxxx

Either a permanent I/O error occurred when reading a program or data file record, or else the record was read but was found to contain an invalid key field. These conditions generally indicate that the data base has been invalidly altered; however, the cause of a permanent I/O error may be a hardware malfunction. If the problem persists, contact the installation system programmer. Condition code = 4.

DIBDB458 SPECIFIED MARG PARAMETER IGNORED FOR FILE xxxxxxxx.

The MARG parameter specified on the WRITE control statement has been ignored, either because the output data set is not a card-image data set (applies to output of CALL-OS data files converted to OS/360 data sets with the FORMDATA option) or because the output is of a type which has standard margins (of 1,72) which may not be overridden (applies to FORTRAN source programs, all object programs, and data files for which FORMDATA has not been specified). Condition code = 0.

DIBDB459 ERROR - MARGINS SPECIFIED TOO CLOSE TOGETHER FOR OUTPUT OF PROGRAM xxxxxxxx

For BASIC programs, the right margin must be greater than or equal to the left margin plus six. Processing of the specified program terminates. Condition code = 4.

DIBDB461 ERROR - FILE xxxxxxxx CONTAINS FORMAT VIOLATION [; REMAINDER OF RECORD CANNOT BE OUTPUT.]

Program or data file xxxxxxxx was found to be in violation of the format required for a CALL-OS file of this type. This problem indicates that the data base has been invalidly altered. The erroneous program or data file may be inspected by using the WRITE function with OUTPUT=PRINT and the HEX option. Condition code = 4.

DIBDB463 DATA FILE xxxxxxxx CONTAINS EMPTY RECORD.

The specified data file contains an empty record. This could happen when a file is defined but never subsequently written into. Processing of the data file terminates. Condition code = 4.

DIBDB465 FILE xxxxxxxx HAS BEEN OUTPUT TO DATA SET SPECIFIED.

When OUTPUT=CARD or OSDS, this message indicates that the WRITE function has been completed successfully. Condition code = 0.

DIBDB475 FILE xxxxxxxx IS PROTECTED.

The specified data file or program cannot be output by the WRITE function because it is protected and being accessed through a *, **, or *** library. Condition code = 4.

DIBDB479 NO FILE FOUND WHICH MET CONTROL CARD SPECIFICATIONS

No output was performed because a file could not be found which satisfied all relevant specifications (NAME, LANG, etc.) given on the utility control statement. Condition code = 4.

DIBDB501 ERROR - BAD OPEN OF TAPEOUT DATA SET; TAPE TERMINATED.

The TAPEOUT data set could not be opened, and output of the backup tape could not begin. Check the JCL to ensure that a TAPEOUT DD statement is present and specified correctly. Correct the JCL and resubmit the function request. Condition code = 8.

DIBDB511 ERROR - PARTITION/REGION NOT LARGE ENOUGH FOR TAPE; FUNCTION TERMINATED.

The TAPE function could not be executed due to insufficient main storage. Specify a larger partition or region for the data base utility and resubmit the function. Condition code = 8.

DIBDB521 ERROR - BACKUP TAPE CANNOT BE WRITTEN FOR A SUB GROUP DIRECTORY ALONE.

A FROMUSER of the form aaan00 was specified and either no FROMUSR2 was specified or FROMUSER and FROMUSR2 are identical. However, a backup tape only contains sub group directory entries when a user number or range of user is specified. Condition code = 8.

DIBDB531 ERROR - INCORRECT PARAMETER SPECIFIED FOR TAPE; FUNCTION TERMINATED.

Either a control statement parameter has been specified with an unallowable value, or an invalid combination of parameters has been given. See the publication _CALL-OS Executive and Utilities Program Description Manual_ for allowable parameters and parameter combinations. Condition code = 8.

DIBDB541 ERROR - FROMUSER SPECIFIED COULD NOT BE FOUND; TAPE TERMINATED.

The FROMUSER parameter on the control statement either specified a user number in a user group which was not available in the specified cluster, or, if the user group was available, the specified user number could not be found in the equivalency file for the group. Condition code = 8.

DIBDB545 ERROR - REQUIRED PASSWORD INCORRECT OR NOT SPECIFIED; TAPE TERMINATED.

The PASSWORD parameter is required on the control statement in order to perform the requested TAPE function; either the PASSWORD parameter was not specified, or the specified password did not match the corresponding password in the equivalency file, or the SYSLIB password is required and the system group in the specified cluster was not available (no SYSGRPnn DD statements were provided). Condition code = 8.

DIBDB551 I/O ERROR OR INCORRECT KEY DETECTED WHEN READING xxx. LINK FOR zzzzzzzzzzz.

Either a permanent I/O error occurred when reading an equivalency file, catalog, or directory record for the FROMUSER specified on the control statement, or else the record was read but was found to contain an invalid key field. In this message:

xxx is EQU, DIR, or CAT

zzzzzzzzzzz is one of the following:

- A user group specification of the form aaazzz when an equivalency record was being read
- USER aaannn when either a catalog or a * directory record was being read
- ** LIBRARY when a ** directory record was being read

These conditions generally indicate that the data base has been invalidly altered; however, the cause of a permanent I/O error may be a hardware malfunction. If the problem persists, contact the installation system programmer. Condition code = 4.

DIBDB569 BACKUP/ARCHIVE TAPE HAS BEEN WRITTEN AS SPECIFIED/REQUIRED.

This is an informative message indicating that the TAPE function has produced the requested tape. Condition code = 0.

DIBDB579 NO FILE FOUND WHICH MET CONTROL CARD SPECIFICATIONS.

No output was performed because a file could not be found which satisfied all relevant specifications (NAME, LANG, etc.) given on the utility control statement. Condition code = 4.

DIBDB601 UNABLE TO OBTAIN MAIN STORAGE FOR BUFFER.

Not enough main storage was available to satisfy a GETMAIN request for storage for a buffer area. Increase task area size and rerun. Condition code = 8.

DIBDB602 ERROR - PASSWORD FOR aaannn DOES NOT MATCH PASSWORD ON CONTROL STATEMENT.

The password in the equivalency entry for the named user does not match the password given on the control statement. Correct PASSWORD parameter and rerun. Condition code = 8.

DIBDB603 NO INPUT PARAMETER; INPUT=CARD ASSUMED.

The INPUT parameter was not given on the control statement; the default of INPUT=CARD was used. Condition code = 0.

DIBDB604 ERROR - INVALID xxxxxxxx PARAMETER SPECIFIED.

The value given for the named parameter is not valid. Correct the parameter and rerun. Condition code = 4 for MARG, LINEGEN, and LINEINC parameters; otherwise, condition code = 8.

DIBDB605 xxxxxxxx PARAMETER SHOULD NOT BE SPECIFIED; PARAMETER IGNORED.

The named parameter should not be specified with the other parameters specified on the control statement. The parameter is ignored. Condition code = 4.

DIBDB606 NO LANG PARAMETER SPECIFIED; LANG=BASIC ASSUMED.

The LANG parameter was not given on the control statement; the default of LANG=BASIC was used. Condition code = 0.

DIBDB607 ERROR - NO xxxxxxxx PARAMETER SPECIFIED.

The named parameter was expected and was not found. Supply the parameter on control statement and rerun. Condition code = 8.

DIBDB608 MARGINS OF nn AND mm ASSUMED.

The margins indicated were used. Condition code = 0.

DIBDB609 NO LINEGEN PARAMETER SPECIFIED; INITIAL LINE NUMBER OF 00100 ASSUMED.

The LINEGEN parameter was not given on the control statement; a default initial line number of 00100 was used. Condition code = 0.

DIBDB610 INVALID LINEGEN PARAMETER; INITIAL LINE NUMBER OF 00100 ASSUMED.

The value given for the LINEGEN parameter is not valid. A default initial line number of 00100 was used. Condition code = 4.

DIBDB611 NO LINEINC PARAMETER SPECIFIED; LINE NUMBER INCREMENT OF 00010 ASSUMED.

The LINEINC parameter was not given on the control statement; a default line number increment of 00010 was used. Condition code = 0.

DIBDB612 INVALID LINEINC PARAMETER; LINE NUMBER INCREMENT OF 00010 ASSUMED.

The value given for the LINEINC parameter is not valid. A default line number increment of 00010 was used. Condition code = 4.

DIBDB613 NO FILE PARAMETER SPECIFIED; FILE=xxxx ASSUMED.

The FILE parameter was not given on the control statement; the specified default (xxxx) was used. Condition code = 0.

DIBDB614 ERROR - USER AND FROMUSER DO NOT SPECIFY THE SAME SUB GROUP.

When a FROMUSER number of the form bbbr00 is given, the first four characters of the USER number must be the same as the first four characters of the FROMUSER number. Correct the USER or FROMUSER parameter and rerun. Condition code = 8.

DIBDB615 ERROR - USER AND USR2 DO NOT SPECIFY THE SAME SUB GROUP.

When a range of users is specified, the first four characters of the USER number must be the same as the first four characters of the USR2 number. Correct USR2 parameter and rerun. Condition code = 8.

DIBDB616 ERROR - USER NUMBER IS NOT LESS THAN OR EQUAL TO USR2 NUMBER.

When a range of users is specified, the USER number must be less than or equal to the USR2 number. Correct either the USER parameter or the USR2 parameter and rerun. Condition code = 8.

DIBDB617 ERROR - LAST TWO DIGITS OF FROMUSER NUMBER DO NOT EQUAL LAST TWO DIGITS OF USER NUMBER.

When a range of users is specified, the last two digits of the FROMUSER number must be the same as the last two digits of the USER number. Correct either the USER number or the FROMUSER number and rerun. Condition code = 8.

DIBDB618 REQUIRED DATA SETS FOR USER xxxxxx NOT OPEN.

The data set for the specified user number was not open. Ensure that a DD statement for the data set is included in the JCL. If the FROMUSER specified is not a directory and a range of users is specified, condition code = 8. If the FROMUSER is a directory and a single file has been specified, condition code = 8; otherwise, condition code = 4.

DIBDB619 FILE=xxxxxxxx NOT FOUND IN aaannn LIBRARY.

The named file to be transferred was not found in the catalog for the specified user number. Ensure that the name of the file to be transferred is correctly given and rerun. Condition code = 8.

DIBDB620 FILE=xxxxxxxx FOR USER aaannn NOT TYPE SPECIFIED BY FILE PARAMETER.

The named file to be transferred is not the same type as specified by the FILE parameter. For example, FILE=DATA was given on the control statement, and the file was found to be a program. Ensure that the correct file type is given on the control statement and rerun. Condition code = 8.

DIBDB621 FILE=xxxxxxxx FOR USER aaannn NOT LANGUAGE SPECIFIED BY LANG PARAMETER.

The named file to be transferred is not the same language type as specified by the LANG parameter. Ensure that the correct language type is given on the control statement and rerun. Condition code = 8.

DIBDB622 FILE=xxxxxxxx PROTECTED.

The named file to be transferred from the ***Library, the **Library, or a *Library is a run-only program and can not be transferred to another user's catalog. Condition code = 4 if multiple files are being processed; Condition code = 8 if a single file is being processed.

DIBDB623 FILE=xxxxxxxx FOR USER aaannn SECURED.

The named file to be transferred has not been released by the specified user. (See CALL-OS Terminal Operations Manual for a description of the RELEASE command.) Condition code = 4 if multiple files are being processed; condition code = 8 if a single file is being processed.

DIBDB624 ERROR - NO EQUIVALENCY ENTRY FOR aaannn NUMBER {ON DIRECTORY MERGE}.

The user number specified was not found in the user group equivalency file; condition code = 8. If directory entries are being merged into a subgroup directory, the message indicates that there is no equivalency entry in the "to" subgroup range for the pooling user number; condition code = 4. In either case, ensure that the USER number is a validated number and rerun.

DIBDB625 UNABLE TO OPEN DATA SET; DDNAME=dddddddd.

The data set defined by the specified DD statement could not be opened. Ensure that the DD statement is present and specified correctly. Condition code = 8.

DIBDB626 LINE nnnnn TRUNCATED TO 237 CHARACTERS.'

The input statement exceeded the maximum allowable length and was truncated to 237 characters. Condition code = 4.

DIBDB627 MAXIMUM xxxxx SIZE EXCEEDED; REMAINING INPUT IGNORED.

When xxxxx is FILE, this message indicates that the input file has reached the maximum allowable number of input records. When xxxxx is LINE#, this message indicates that the line number generated has exceeded the maximum of five characters. In either case, the remaining input records are ignored; condition code = 4.

DIBDB628 NO INPUT FILE FOR THIS CONTROL STATEMENT.

There was no input deck following the control statement when INPUT=CARD was specified. Condition code = 4.

DIBDB629 xxxxx I/O ERROR FOR USER aaannn, CLUSTER k;
RECORD=xxxxxxxxxxx; DTTR=xxxxxxxx.

An I/O error of the specified type occurred. The user number, cluster, record type (or file name), and DTTR are given. For multiple users, condition code = 4; for a single user and multiple files (RECORD=filename or FILE DESCPT), condition code = 4; otherwise condition code = 8.

DIBDB630 INCORRECT DATA SET FORMAT; DDNAME=dddddddd.

The data set defined on the specified DD statement did not have the required format as follows:

1. For either program input or data file input without the FORMDATA option:

   RECFM=F,BLKSIZE=80 or

   RECFM=FB,LRECL=80

2. For data file input with the FORMDATA option:

   RECFM=F,BLKSIZE less than or equal to 225, or

   RECFM=FB,LRECL less than or equal to 255, or

   RECFM=V or VB,LRECL less than or equal to 259

For either case, condition code = 8.

DIBDB631 ALL FILES PROCESSED FOR THIS CONTROL STATEMENT.

Indicates end of processing for the current control statement. Condition code = 0.

DIBDB632 I/O ERROR READING aaannn EQUIVALENCY RECORD.

An I/O error occurred while attempting to read the equivalency record for the specified user. Processing of the control statement is terminated. Condition code = 8.

DIBDB633 MARGIN PARAMETER MAY NOT BE OVERRIDDEN.

The MARG parameter was specified when standard margins must be used for the input. The MARG parameter is ignored. Condition code = 4.

DIBDB634 ERROR - UNABLE TO FIND OSDS MEMBER xxxxxxxx.

The named member was not found in the partitioned data set defined by the LIBRARY DD statement. For a single file, condition code = 8; for multiple files, condition code = 4.

DIBDB635 ABOVE RECORD HAS INVALID LINE NUMBER.

For INPUT=CARD or OSDS and LANG=BASIC, a line number was found to be either not all numeric characters or not sequential. The record is ignored. Condition code = 4.

DIBDB636 ABOVE RECORD NOT IN SEQUENCE.

OPTICNS=SEQ was specified on the control statement and the record printed was not in sequence. The record is ignored. Condition code = 4.

DIBDB637 ERROR - INPUT DATA COUNT DOES NOT EQUAL COUNT INDICATED ON /$ RECORD.

For INPUT=CARD or OSDS and FILE=DATA without the FORMDATA option, an input record was found to have a data count that did not equal the amount of data actually present for the record. Condition code = 8.

DIBDB638 ERROR - NO FILE WAS FOUND WHICH MET CONTROL STATEMENT REQUIREMENTS.

For INPUT=OSDS, TAPE, or DISK, no file was found which exactly met those requirements specified on the control statement. Condition code = 4.

DIBDB639 FILE=xxxxxxxx NOT POOLED.

The named file was to be transferred from a shared library; the file could not be found. For multiple files, condition code = 4; for a single file, condition code = 8.

DIBDB640 FUNCTION TERMINATED.

The INSERT/REPLACE operation requested has been terminated. The condition code is determined by the severity of the error which caused termination.

DIBDB641 FILE xxxxxxxx TO BE INSERTED ALREADY IN aaannn LIBRARY.

On a search of the specified user's catalog for an INSERT function, the named file was found to already exist in the catalog. The input file was not inserted. Condition code = 4 if multiple files are being inserted; condition code = 8 if a single file is being inserted.

DIBDB642 FILE xxxxxxxx TO BE REPLACED IN aaannn LIBRARY NOT SAME FILE TYPE AS INPUT FILE.

The named file to be replaced was found to be a different file type from the input file specified by the FILE parameter. The existing file was not replaced by the input file. Either ensure that the FILE parameter is correct or purge the active file in the catalog and insert the new file. Condition code = 4 if multiple files are being transferred; condition code = 8 if a single file is being entered.

DIBDB643 FILE xxxxxxxx TO BE REPLACED IN aaannn LIBRARY IS LOCKED.

The named file to be replaced was found to have the LOCK attribute and OPTIONS=(UNLOCK) was not specified. The existing file was not replaced by the input file. Condition code = 4 if multiple files are being transferred; condition code = 8 if a single file is being processed.

DIBDB644 NO DISK SPACE AVAILABLE FOR CATALOG/DIRECTORY LINK FOR xxxxxx LIBRARY.

A request was issued for a half-track record needed for a catalog/directory link in the specified library; the response indicated no more disk space was available in the group data set(s). The group should be reorganized to return purged space to a usable state. Condition code = 8.

DIBDB645 NO DISK SPACE AVAILABLE FOR FILE xxxxxxxx IN aaannn LIBRARY.

A request for disk space needed for writing the named file indicated none was available in the group data set(s). The group should be reorganized to return purged space to a usable state. Condition code = 8.

DIBDB646 FILE xxxxxxxx TO BE REPLACED DID NOT EXIST PREVIOUSLY IN aaannn LIBRARY.

No file of the name given existed in the data base before the requested REPLACE operation. The input file was inserted into the USER library. Condition code = 0.

DIBDB647 FILE xxxxxxxx SAVED IN aaannn LIBRARY.

The file named has been inserted or replaced in the library for the specified user. Condition code = 0.

DIBDB648 FILE xxxxxxxx ALREADY POOLED IN yy LIBRARY.

An attempt to pool the named file indicated an entry already existed with that filename in the specified library. Condition code = 4.

DIBDB649 FILE xxxxxxxx POOLED IN yy LIBRARY.

The file named has been pooled in the library specified. Condition code = 0.

DIBDB650 ERROR - FORMDATA OPTION IS INVALID.

The FORMDATA option implies that an external data file is being input. However, either LANG=BASIC or OPTIONS=OBJECT has been specified, both of which are inconsistent with OPTIONS=FORMDATA. Condition code = 8.

DIBDB651 ERROR - /$ RECORD ENCOUNTERED AFTER LAST INPUT DATA RECORD HAS BEEN READ.

For INPUT=OSDS or CARD and FILE=DATA without the FORMDATA option, a /$ control record has been read after the previous /$ record indicated that it was the last record for the input file. Condition code = 8.

DIBDB652 ERROR - NO /$ RECORD FOR INPUT FILE.

A request was made to input a data file without OPTIONS=FORMDATA from CARD or OSDS input. The input was not preceded by a /$ control record. Condition code = 8.

DIBDB653 ERROR - /$ RECORD ENCOUNTERED READING STORED OBJECT CODE INPUT.

For INPUT=OSDS or CARD and OPTIONS=OBJECT, a /$ control record was found after the first input record. Condition code = 8.

DIBDB654 ERROR - INVALID /$ RECORD FOUND.

While attempting to input a stored object code program or a data file without the FORMDATA option, the /$ control record was found to contain an invalid field. Condition code = 8.

DIBDB655 FILE xxxxxxxx MERGED IN yyyyyy LIBRARY.

An entry from the directory specified by the FROMUSER parameter on the INSERT/REPLACE control statement was inserted or replaced in the sub group directory (when yyyyyy is of the form aaan00) or the **directory (when yyyyyy is **LIB). Condition code = 0.

DIBDB671 ERROR - BAD OPEN OF TAPEIN DATA SET; INSERT/REPLACE TERMINATED.

INPUT=TAPE was specified on the function statement but the TAPEIN data set could not be opened. Check the JCL to ensure that a TAPEIN DD statement is present and is specified correctly. Correct the JCL and resubmit the function request. Condition code = 8.

DIBDB701 ERROR - BAD OPEN OF TAPEIN DATA SET; RECON TERMINATED.

The TAPEIN data set could not be opened and reconstruction cannot begin. Check the JCL to ensure that a TAPEIN DD statement is present and is specified correctly. Correct the JCL and resubmit the function request. Condition code = 8.

DIBDB711 ERROR - PARTITION/REGION NOT LARGE ENOUGH FOR RECON; FUNCTION TERMINATED.

The requested conversion cannot be completed due to lack of main storage. The function should be resubmitted with a larger partition or region specified for the data base utility. Condition code = 8.

DIBDB721 ERROR - RANGE CARDS MUST BEGIN WITH '/$'. INVALID CARD(S) SKIPPED.

This message appears while the range cards are being read. One or more cards following the RECONSTRUCT function statement do not have /$ as the first two characters and another ./ function statement has not been encountered. The invalid cards are skipped. All remaining range cards are processed to detect other errors and a sorted listing of the ranges is printed. The listing is followed by message DIBDB739. Condition code = 8.

DIBDB722 ERROR - RANGE FIELD NOT PRECEDED BY BLANK. INVALID FIELD(S) SKIPPED.

This message appears while the range cards are being read. A format error has been detected during the scan of the range card: the character in column 3, 16, 29, 55, or 68 was not a blank. All remaining range cards are processed to detect other errors and a sorted listing of the ranges is printed. The listing is followed by message DIBDB739. Condition code = 8.

DIBDB725 ERROR - TOO MANY ENTRIES FOR THE RANGE TABLE. MAXIMUM IS 200.

The maximum number of ranges that may be specified on range cards is 200. Either reorganize the ranges into a smaller number or reassemble the RECONSTRUCT function module with a larger range table. A sorted listing of the first 200 ranges is printed. This listing is followed by message DIBDB739. Condition code = 8.

DIBDB731 ERROR - RANGE 'aaammmbbbnnn' HAS INVALID CHARACTER(S).

This message appears while the range table is being sorted. The indicated range does not follow the required format of aaammmbbbnnn where a and b are alphabetic characters (A through Z) and m and n are numeric characters (0 through 9). All remaining ranges are processed to detect other errors and a sorted listing of the ranges is printed. This listing is followed by message DIBDB739. Condition code = 8.

DIBDB732 ERROR - RANGE 'aaammmbbbnnn': LAST IS LOWER THAN FIRST.

This message appears while the range table is being sorted. For the indicated range, the high order user number (last six characters of the range) precedes the low order user number (first six characters) in the collating sequence. All remaining ranges are processed to detect other errors and a sorted listing of the ranges is printed. This listing is followed by message DIBDB739. Condition code = 8.

DIBDB733 ERROR - RANGE 'aaammmbbbnnn' OVERLAPS ANOTHER RANGE.

This message appears while the range table is being sorted. User numbers in the indicated range overlap the user numbers of another range. All remaining ranges are processed to detect other errors and a sorted listing of the ranges is printed. This listing is followed by message DIBDB739. Condition code = 8.

DIBDB734 ERROR - RANGE 'aaammmbbbnnn' IS NOT WITHIN SPECIFIED USRGROUP.

This message appears while the range table is being sorted. All or some of the user numbers in the indicated range fall outside the user number limits of the requested user group. All remaining ranges are processed to detect other errors and a sorted listing of the ranges is printed. This listing is followed by message DIBDB739. Condition code = 8.

DIBDB739 RECON TERMINATED. CORRECT RANGE CARDS BEFORE RE-RUNNING.

This message follows the sorted range listing when one or more errors have been detected, either while reading the range cards or sorting the range table. Specific errors messages precede the listing. The requested conversion is not performed and the next function statement is processed. Condition code = 8.

DIBDB750 ERROR - SOURCE PROGRAM TOO LONG; SIZE IN CATALOG CHANGED TO 28848 BYTES: USER aaammm, FILENAME xxxxxxxx

A catalog entry from a CALL/360: Standalone system backup tape contains a length specification larger than the maximum allowed for CALL-OS (28,848 bytes). The catalog entry is set to the allowable length. This message is followed by message DIBDB751 when the record associated with the catalog entry is read. Condition code = 4.

DIBDB751 ERROR - SOURCE PROGRAM LENGTH GREATER THAN CATALOG SAYS IT IS; TRUNCATED TO FIT ON ALLOCATED DISK RECORD(S).

A source program is longer than the catalog entry specifies. This message will usually follow message DIBDB750 and indicates that the record associated with the reset catalog entry has just been read. If message DIBDB750 does not appear and the backup tape is not from the CALL/360: Standalone System, a program error has occurred. In either case, the input program is truncated and a dump of the entire program is printed. Condition code = 4.

DIBDB755 ERROR - OBJECT PROGRAM OR DATA FILE TOO LONG; SIZE IN CATALOG INCREASED: USER aaammm, FILENAME xxxxxxxx

An object program or data file is longer than the catalog entry indicates. The length in the catalog is increased to correspond to the actual length. Condition code = 4.

DIBDB756 ERROR - OBJECT PROGRAM OR DATA FILE TOO SHORT; SIZE IN CATALOG DECREASED: USER aaammm, FILENAME xxxxxxxx

An object program or data file is shorter than the catalog entry indicates. The length in the catalog is decreased to correspond to the actual length. Condition code = 4.

DIBDB761 ERROR - INVALID RECORD TYPE; RECORD SKIPPED.

The type code of the record read from the CALL/360: Standalone System backup tape cannot be recognized. The record is ignored. Condition code = 4.

DIBDB762 ERROR - CATALOG AND FILE TYPE DISAGREE; RECORD SKIPPED.

The type code in the catalog entry does not match the type code of the record to which the catalog entry points. The record is not moved into the data base. Condition code = 4.

DIBDB765 ERROR - FILENAME NOT IN CATALOG; FILE DROPPED.

The catalog does not contain an entry for the specified file. The file is not moved into the data base. Condition code = 4.

DIBDB768 ERROR - TAPE DOES NOT MATCH CATALOG; RECORD SKIPPED.

A record has been read from a backup tape but a catalog entry cannot be found which points to the record. The record is not moved into the data base. Condition code = 4.

DIBDB773 ERROR - MISSING *** CATALOG; RECON TERMINATED.

A system group program or data file was read but its catalog has not been read. Conversion cannot continue. Condition code = 8.

DIBDB775 ERROR - MISSING EQUIVALENCY ENTRY; RECON TERMINATED.

No equivalency entry was found in the backup tape for a discovered user number. An equivalency record or at least an entry is missing. Conversion cannot continue. Condition code = 8.

DIBDB781 ERROR - OUT OF DISK SPACE; RECON TERMINATED.

All the space allocated to the user group has been used. Before the conversion may be attempted again, UTILX must be run to delete this user group from the CALL-OS index and U#UTIL1 must then be run to allocate more space to the user group. A larger allocation is achieved by either adding a data set to the group which was too small or scratching the data sets in the group with IEHPROGM and allocating more space on the DD statements supplied to U#UTIL1. Condition code = 8.

DIBDB791 PERMANENT DISK I/O ERROR; RECON TERMINATED.

A permanent read or write error has occurred. The disk pack or drive should be checked. It may be necessary to flag bad tracks and assign alternates. The unit number of the device in error appears on the OS/360 system operator's console. Condition code = 8.

DIBDB801 ERROR - BAD OPEN OF dddddd DATA SET; ACCOUNT TERMINATED.

The data set defined by the specified DD statement (either TAPEJ or PRINTJ) could not be opened and the requested ACCOUNT function cannot be performed. Check the JCL to ensure that the DD statement is present and is specified correctly. Correct the JCL and resubmit the function request. Condition code = 8.

DIBDB811 ERROR - PARTITION/REGION NOT LARGE ENOUGH FOR ACCOUNT; FUNCTION TERMINATED.

The requested ACCOUNT function cannot be performed due to lack of main storage. The function should be resubmitted with a larger partition or region specified for the data base utility. Condition code = 8.

DIBDB831 ERROR - INCORRECT PARAMETER SPECIFIED FOR ACCOUNT; FUNCTION TERMINATED.

The OPTIONS parameter has been specified with an unallowable value. See the publication <u>CALL-OS Executive and Utilities Program Description Manual</u> for a description of the allowable parameter values and parameter combinations. Condition code = 8.

DIBDB845 ERROR - REQUIRED PASSWORD INCORRECT OR NOT SPECIFIED; ACCOUNT TERMINATED.

The PASSWORD parameter must be specified on the utility control statement for the ACCOUNT function. Either the specified password does not match the corresponding password in the equivalency file, or the PASSWORD parameter is missing, or the SYSLIB password was specified and the system group data set(s) in the specified cluster was not available (no SYSGRPnn DD statements were provided). Condition code = 8.

DIBDB851 ERROR - UNABLE TO READ FULL TRACK EQU; ACCOUNT TERMINATED

A disk I/O error occurred while trying to read the equivalency file for the user group. Condition code = 8.

DIBDB852 ERROR - RECORD READ NOT EQU; ACCOUNT TERMINATED.

The record read using an equivalency file address was not an equivalency file. Condition code = 8.

DIBDB853 ERROR - UNABLE TO READ CATALOG/DIRECTORY RECORD; ACCOUNT TERMINATED.

A disk I/O error occurred while trying to read a catalog or directory record of a user of this group. Condition code = 8.

DIBDB854 ERROR - RECORD READ NOT CATALOG.

The record read using a catalog address was not a catalog. Condition code = 4.

DIBDB855 ERROR - RECORD READ NOT DIRECTORY.

The record read using a directory address was not a directory. Condition code = 4.

DIBDB856 ERROR - INCORRECT USER NUMBER THIS CATALOG/DIRECTORY RECORD.

The user number in the catalog or directory did not match the user number in the equivalency record. Condition code = 4.

DIBDB861 ERROR - NO EOD THIS EQU RECORD.

No end of data (X'01') was found for the equivalency file of the user group; however, it is inserted. Condition code = 4.

DIBDB863 ERROR - UNABLE TO WRITE EQU RECORD TO DISK.

A disk I/O error occurred while trying to rewrite an equivalency record of the user group. Condition code = 4.

DIBDB901 ERROR - INVALID PASSWORD.

Either no password was specified, or the specified password contained leading blanks, or the specified password did not match the password in the SYSLIB equivalency record. Condition code = 8.

DIBDB902 ERROR - NO UGT FOR USRGROUP aaabbb.

No entry was found in the user group data set translation table for the specified user group. Condition code = 8.

DIBDB903 ERROR - xxxxxx FAIL OF USER yyyyyy zzzz RECORD.

An I/O error occurred during a read or write of either a user or SYSLIB equivalency record, or a user catalog record. Partial execution of the JOBFIND function may have been completed. Resubmit control statement. Condition code = 8.

DIBDB904 ERROR - OPEN FAIL OF COBI INDEX

The COBI index data set could not be opened. Ensure that the DD statement is present and specified correctly. Condition code = 8.

DIBDB905 ERROR - NO SYSLIB EQUIVALENCY RECORD.

An equivalency record for SYSLIB could not be found in the SYSGRP data set. This is a system error which must be corrected before the JOBFIND function can be used. Condition code = 8.

DIBDB906 USRGROUP aaabbb CATALOGS UPDATED.

The catalogs for all users within the specified user group were updated according to associated COBI job entries in the COBI index. Condition code = 0.

DIBDB907 ERROR - GETMAIN FOR nnnn BYTES FAILED.

Core storage in the amount specified was not available for buffer use. Condition code = 8.

DIBDB908 ERROR - NO DISK SPACE AVAILABLE.

Disk storage could not be allocated for a newly-generated catalog record. Condition code = 8.

DIBDB909 ERROR - NO EQUIVALENCY ENTRY FOR USER aaannn.

The user specified in the COBI index record is not a validated user in its user group equivalency file. Condition code = 4.

DIBUT202 "ERRNO" NOT BETWEEN 220 AND 229.

The ERRNO parameter supplied is invalid. The value must be between 220 and 229. Correct the parameter and restart the job.

DIBUT203 "SVC" NUMBER NOT BETWEEN 200 AND 225.

The SVC number supplied is invalid. The value must be between 200 and 225. Correct the parameter and restart the job.

DIBUT204 "NUC" FIELD ERROR.

The NUC parameter is invalid. Both the from and to nucleus values must be between 1 and 9. Correct the parameter and restart the job.

DIBUT205 "TYPE" FIELD ERROR.

The TYPE parameter is invalid. The value must be either MFT or MVT only. Correct the parameter and restart the job.

DIBUT206 "COBI" FIELD ERROR.

The COBI parameter is invalid. The value must be either YES or NO. Correct the parameter and restart the job.

DIBUT500 MORE THAN ONE EXTENT IS NOT ALLOWED FOR DDNAME dddddddd

An old data set was defined on the specified DD statement (dddddddd) and this data set contained more than one extent. A user ABEND 507 is issued and processing terminates. Allocate sufficient space in one extent for the data set.

DIBUT501 CONTIGUOUS EXTENT WAS NOT REQUESTED FOR DDNAME dddddddd

The specified DD statement (dddddddd) did not contain the CONTIG subparameter in the SPACE parameter. A user ABEND 507 is issued and processing terminates. Correct the SPACE parameter and resubmit the job.

DIBUT502 SECONDARY ALLOCATION SHOULD NOT BE SPECIFIED FOR DDNAME dddddddd

The specified DD statement (dddddddd) contains a secondary allocation in the SPACE parameter. A user ABEND 507 is issued and processing terminates. Correct the SPACE parameter and resubmit the job.

DIBUT503 DATA SET FOR DDNAME dddddddd FOUND TO BE NOT VACUOUS

The data set defined by the specified DD statement (dddddddd) contains one or more records but the utility only formats data sets which are completely empty. A user ABEND 507 is issued and processing terminates. Scratch the data set and ensure that it is empty; resubmit the job.

DIBUT504 BAD OPEN FOR DATA SET, DDNAME dddddddd

It was impossible to open dddddddd. A user ABEND 507 is issued and processing terminates. Either supply the missing DD statement or correct the existing one; resubmit the job. If the DD statement is correct, notify the installation system programmer immediately.

DIBUT507 CYLINDER ALLOCATION NOT SPECIFIED FOR DDNAME CBSYSINx

The DD statement for the specified COBI input data set (where x is either A or B) did not contain CYL specification in the SPACE parameter. A user ABEND 507 is issued and processing terminates. Correct the SPACE parameter and resubmit the job.

DIBUT508 UNEQUAL SPACE ALLOCATION FOR CBSYSINA AND CBSYSINB

The two COBI input data sets must be identical. A user ABEND 507 is issued and processing terminates. Check the DD statements for both data sets to ensure that CYL and CONTIG are specified in the SPACE parameter for both and that the number of cylinders specified is the same for both.

DIBUT509 THE U#5xxxxx UTILITY FUNCTION HAS COMPLETED SUCCESSFULLY

This is an informative message, indicating that the specified utility function (where xxxxx is INIT, CBXPN, RINIT, or PURGE) has been completed successfully. Condition code = 0 unless message DIBUT529 is issued during execution of U#5PURGE; in this case, condition code = 8.

DIBUT517 INVALID PARAMETER SPECIFIED ON THE EXEC STATEMENT

A parameter on the EXEC statement could not be recognized. Ensure that all parameters are spelled correctly. A user ABEND 507 is issued and processing terminates.

DIBUT518 NUMERIC VALUE SPECIFIED FOR A PARAMETER IS TOO LONG

The value specified for a parameter has more than four numeric digits. A user ABEND 507 is issued and processing terminates.

DIBUT519 INVALID CHARACTER WITHIN OR ENDING A NUMERIC VALUE

The value specified for a parameter either contains a nonnumeric character within it or, for all but the last one in the parameter list, ends with a character other than a comma. A user ABEND 507 is issued and processing terminates.

DIBUT520 DUPLICATE PARAMETERS SPECIFIED ON EXEC STATEMENT

The EXEC statement contains two or more duplicate parameters. A user ABEND 507 is issued and processing terminates.

DIBUT522 ZERO VALUE FOUND FOR ONE OR MORE SPECIFIED PARAMETERS

A zero was found when a numeric parameter value was expected. A user ABEND 507 is issued and processing terminates.

DIBUT524 COBI JCL DATA SET 'CBJCL' DOES NOT START ON A CYLINDER BOUNDARY

For U#5INIT and U#5RINIT, the space allocated to the COBI JCL data set must start on a cylinder boundary. A user ABEND 507 is issued and processing terminates. Correct the SPACE parameter on the CBJCL DD statement and restart the job.

DIBUT525 THE PARM VALUE SPECIFIED FOR CBRDSET IS OUT OF THE RANGE 1 THRU 4

For U#5INIT or U#5CBXPN, the value specified for the CBRDSET parameter was outside the range from 1 through 4. A user ABEND 507 is issued and processing terminates. Correct the CBRDSET parameter value and restart the job.

DIBUT526 CURRENT COBI INDEX WITH DDNAME dddddddd IS NOT IN PROPER FORMAT

During execution of U#5CBXPN or U#5PURGE, the current COBI index was found to have an incorrect format. A user ABEND 507 is issued and processing terminates. For U#5CBXPN, either the data set defined on the CBOLDX DD statement is not the COBI index, or, if it is the COBI index, the information in it has been altered. In the first case, ensure that the CBOLDX DD statement defines the current COBI index; in the second case, the expand function should not be executed because it copies the current COBI index before expanding it. For the second case of U#5CBXPN and U#5PURGE, use the reinitialization function (U#5RINIT) to reinitialize all COBI data sets.

DIBUT527 CURRENT COBI JCL DATA SET WITH DDNAME CBOLDJCL IS NOT IN PROPER FORMAT.

The CBOLDJCL DD statement does not define a properly formatted COBI JCL data set. A user ABEND 507 is issued and processing terminates. Ensure that the DD statement is specified correctly. If it is, the U#5INIT utility must be executed to format the JCL data set.

DIBUT528 DUPLICATE DSNAME ON VOL=SER=nnnnnn

After the COBI index has been expanded successfully, the U#5CBXPN function switches the data set names on the CBOLDX and CBNDX DD statements. When the current and expanded indeces are on different volumes, this message indicates that a duplicate data set name was encountered while the switch was being made. For example, the user-specified data set name on the CBNDX DD statement is already present on the volume which contains the current index (defined with the CBOLDX DD statement). A user ABEND 507 is issued and processing terminates.

DIBUT529 SOLID I/O ERROR IN VTOC ON VOL=SER=nnnnnn

When this message occurs during execution of U#5CBXPN, it indicates that the switch of data set names failed due to a permanent I/O error during a RENAME operation; user ABEND 507 is issued and processing terminates. When this message occurs during execution of U#5PURGE, it indicates that a permanent I/O error occurred during a SCRATCH operation on the volume table of contents for a volume defined on a SCANxx DD statement; condition code = 8 is posted, and processing continues.

DIBUT530 NUMERIC VALUE SPECIFIED FOR A PARAMETER IS TOO LARGE

The binary value of a parameter specified on the EXEC statement exceeds 32,000. A user ABEND 507 is issued and processing terminates.

DIBUT531 NO SCANXX DD CARDS PRESENT. U#5PURGE CANCELED BY OPERATOR

A WTO message (DIBUT510) is issued on the OS/360 system operator's console if there are no scannable data sets to be processed by U#5PURGE (that is, no SCANxx DD statements were included in the JCL). The operator may either continue processing without the data sets or cancel further processing. The above message is issued when the operator has canceled the job. A user ABEND 507 is issued and processing terminates. Supply the missing SCANxx DD statements and restart the job.

DIBUT532 INPUT DATA IN SYSINA AND/OR SYSINB - xxxxxxxx CANCELED BY OPERATOR

A WTO message (DIBUT511) is issued on the OS/360 system operator's console if U#5RINIT or U#5PURGE does not find an EOF record at the beginning of a COBI input data set. The operator may either continue processing or cancel the job. The above message is issued when the operator has canceled the job. A user ABEND 507 is issued and processing terminates. It is possible that either or both of the data sets must be read by the COBI reader program before reinitialization or purging may take place.

DIBUT533 JCL FOR COBI JOB #nnnnn (USER aaannn) HAS BEEN TRUNCATED TO FIT IN NEW JCL FILE.

The number of records per JCL record set has been decreased during execution of U#5CBXPN. The maximum amount of storage for the JCL for any one job is three record sets. The JCL for the specified job submitted by the indicated user has exceeded three record sets. Processing continues. Condition code = 0.

DIBUT534 MORE THAN 100 VOLUMES SPECIFIED WITH SCANxx DD STATEMENTS

For U#5PURGE, no more than 100 SCANxx DD statements may be specified. A user ABEND 507 is issued and processing terminates.

DIBUT535 CBSYSINA/B BLOCK SIZE NOT EQUAL

The block sizes specified for both CBSYSINA and CBSYSINB must be identical; in addition, an equal amount of contiguous space must be allocated in cylinders for both data sets. A user ABEND 507 is issued and processing terminates.

DIBUT536 TOO FEW JCL RECORDS SPECIFIED.

The total number of records in the JCL data set must be at least twice the number of records specified for a record set; that is, at least two record sets must be available. This is because the first record set is used for control information and cannot be used for JCL storage. A user ABEND 507 is issued and processing terminates. Correct the value and restart the utility.

DIBUT537 TOO FEW COBI INDEX RECORDS SPECIFIED.

The COBI index must have at least two records. This is because the first record is used for data set identification and cannot be used for a submitted job. A user ABEND 507 is issued and processing terminates. Correct the value and restart the utility.

DIBUT538 CBSYSINA/B BLOCK SIZE IS OUTSIDE THE PERMISSIBLE RANGE.

The blocksize for the CBSYSINA and CBSYSINB data sets must be within the range from 400 bytes through 3200. A user ABEND 507 is issued and processing terminates. Correct the DCB parameter(s) and restart the utility.

DIBUT539 NEW COBI INDEX IS TOO SMALL FOR CURRENT JOB FILES.

During execution of U#5CBXPN, an attempt was made to contract the COBI index. The new index is too small to contain all of the information from the old index, which is copied sequentially into the new data set. A user ABEND 507 is issued and processing terminates.

DIBUT540 xxxxxxxx DATA SET NOT PREVIOUSLY CATALOGED OR REQUIRED CATALOG INACCESSIBLE.

U#5CBXPN assumes that the present (old) COBI index (when xxxxxxxx is CBOLDX), or the present (old) JCL data set (when xxxxxxxx is CBOLDJCL) was cataloged at the time it was created and tries to uncatalog it during the expand process. However, the old data set could not be uncataloged, either because it was not previously cataloged or because the volume which contains the data set is not mounted.

DIBUT541 SOLID I/O ERROR DETECTED WHILE PROCESSING THE SYSTEM CATALOG.

A solid (unrecoverable) I/O error was detected on the catalog volume while trying to uncatalog the old COBI index during execution of U#5CBXPN. A user ABEND 507 is issued and processing terminates.

DIBUT542 NEW COBI JCL DATA SET IS TOO SMALL FOR CURRENT JOB FILES.

During execution of U#5CBXPN, an attempt was made to contract the COBI JCL data set. The new data set is too small to contain all the JCL from the old data set. A user ABEND 507 is issued and processing terminates.

DIBUT602 OPEN FAILED FOR DDNAME dddddddd

The DIBCONPR utility could not open the SYSLIB, SYSPUNCH, or SYSIN data set; execution terminates. Either supply the missing DD statement or correct the DD statement, if present. If the DD statement is present and is correct, notify the installation system programmer. Condition code = 8.

DIBUT603 DIBCONPR ALLOCATED INSUFFICIENT MAIN STORAGE.

DIBCONPR could not obtain the storage necessary for buffers for reading the procedure library to be converted. Execution terminates. The amount needed for buffers is equal to twice the blocksize of the procedure library. Condition code = 8.

DIBUT607 SYSIN DATA SET IS VACUOUS. NO PROCEDURE NAME CARDS FOUND.

The SYSIN data set did not contain the names of any procedures to be converted. Execution terminates. Condition code = 4.

DIBUT608 DUPLICATE SYSIN CONTROL CARD FOUND FOR THE PROCEDURE NAMED: pppppppp

The SYSIN data set contained two control cards with the same procedure name. The indicated procedure is converted once; the second control card is ignored. Condition code = 4.

DIBUT609 SYSIN CONTAINS TOO MANY CARDS. pppppppp AND ANY FOLLOWING PROCEDURES ARE IGNORED.

The SYSIN data set contained more than 100 control cards. The first 100 procedure names are sorted and the procedures are converted. Starting with the specified procedure name, any subsequent procedure names are ignored. Condition code = 4.

DIBUT611 SOLID IO ERROR WHILE SEARCHING SYSLIB PDS DIRECTORIES.

A permanent I/O error was detected while searching the directory of the SYSLIB data set. Execution terminates. Condition code = 8.

DIBUT612 BLDL FAILED FOR PROCEDURE pppppppp

The specified procedure could not be found in the directory for the SYSLIB data set. Conversion continues with the next procedure. Condition code = 4.

DIBUT613 SYNTAX ERROR DISCOVERED IN THE FOLLOWING PROCEDURE STATEMENT:

During conversion of a cataloged procedure, DIBCONPR detected a syntax error. All statements in the procedure up to the one in error and including all converted statements are printed preceding the message; the statement in which the error was detected is printed following the message. Conversion of this procedure terminates; processing continues with the next procedure. Condition code = 4.

DIBUT614 PROCEDURE pppppppp YIELDS TOO MANY ENTRIES FOR THE xxxxxxxxx STACK.

The specified cataloged procedure is too large for conversion; in this message, xxxxxxxxx is one of the following:

PROCEDURE, the stack which contains the PROC statement, if any, and its continuation cards, including the default values for the symbolic parameters added by DIBCONPR; this stack is limited to 20 cards.

MAIN, the stack which contains all the EXEC and DD statements in the procedure and their continuation cards; this includes all converted statements as well as those which required no conversion; this stack is limited to 200 cards.

TEMPORARY, the stack which is used temporarily during the conversion of each DD statement; this stack is limited to 5 cards.

Following the error message, the contents of the stack which caused the message is printed. Conversion of this procedure terminates; processing continues with the next procedure. Condition code = 4.

DIBUT617 INVALID PARAMETER SPECIFIED ON THE EXEC STATEMENT.

The parameter field of the EXEC statement for DIBCONPR can contain only the SPACE, OSCLASS, and CBCLASS parameters. No other parameters may appear. Either a parameter is misspelled or a parameter other than these three is specified. Execution terminates. Correct the parameter field and restart the job. Condition code = 8.

DIBUT619 PARAMETER VALUE SPECIFIED ON THE EXEC STATEMENT IS INVALID OR TOO LONG.

The OSCLASS and CBCLASS parameters must specify valid OS/360 output classes. The SPACE parameter must specify a valid space allocation value in 50 characters or less. Execution terminates. Correct the parameter in error and restart the job. Condition code = 8.

DIBUT620 DUPLICATE PARAMETERS SPECIFIED ON EXEC STATEMENT.

The parameter field of the EXEC statement for DIBCONPR contains two identical parameters. Execution terminates. Correct the parameter field and restart the job. Condition code = 8.

DIBUT650 HIGHEST CONDITION CODE WAS c.

This is an informative message issued before DIBCONPR terminates execution. It indicates the highest condition code encountered during execution. A condition code of 0 indicates successful execution.

DIBUT651 END OF CALL-OS BATCH INTERFACE UTILITY RUN.

This is an informative message issued when DIBCONPR terminates execution.

*** DSNAME MATCHES DSNAME IN INDEX ***

This message is issued by U#UTIL1; it indicates that the dsname specified on the ddcard matches a dsname on an entry in the index. This diagnostic does not appear on a COMPILER run if the index entry with the matching dsname is a compiler entry with a ddname which matches the ddcard's ddname.

*** FIELD n INVALID - NO UPDATE ATTEMPTED ***

This message is issued by UTILX; it means that the indicated field on the previous detail card has a syntax error, or contains information which is not of the correct form, or is inconsistent with the other card fields or with the present index entries.

*** FORMATTING - BAD OPEN ***

This message is issued by U#UTIL1 during a SYSGROUP or USRGROUP run; the message indicates that OS/360 was unable to open for output the specified data set in order to format it.

*** FORMATTING - DATA SET TOO SMALL ***

This message is issued by U#UTIL1; it can occur on a SYSGROUP or USRGROUP run if the data set specified is the first data set in its group, and it does not have at least three or two tracks, respectively.

*** FORMATTING - I/O ERROR IN WRITING TRACKS ***

This message is issued by U#UTIL1; it indicates that a permanent I/O error occurred while attempting to format a system group or user group data set. The disk pack is possibly bad.

*** FORM OF DDNAME IS NOT CORRECT ***

This message is issued by U#UTIL1. During a WORKSWAP run, this indicates that the first four characters of the ddname on the ddcard being processed are not SWAP, or that the fifth and sixth characters are not decimal digits, or that the ddname is longer than six nonblank characters. During a SYSGROUP run, it indicates that the first six characters are not SYSGRP or the last two characters are not decimal digits. During a USRGROUP run, it indicates that the first six characters are not alphabetic or the last two are not decimal digits. During an OVERLAY run, it indicates that the ddname is not OVLY.

*** INDEX FULL ***

This message is issued by U#UTIL1; it indicates that the CALL-OS index data set is full and hence the data set whose ddname appears above this message cannot be added to the CALL-OS data base. UTILX may be used to delete one or more entries from the index in order to make room for new entries.

*** INVALID USER GROUP - FIRST 3 > SECOND 3 ***

This message is issued by U#UTIL1 and may occur during a USRGROUP run. It indicates that the ddname (on the ddcard being processed) does not represent a valid user group, since the first three characters are greater (in collating sequence) than the second three.

*** INVALID USER GROUP - OVERLAPS USER GROUP IN INDEX ***

This message is issued by U#UTIL1 and may occur during a USRGROUP run. It indicates that the ddname on the ddcard being processed represents an invalid user group, since its range of user ID characters overlaps the range of a user group in the same cluster in the index. The index entry information for the conflicting entry is printed above this diagnostic.

*** MATCHING DDNAME IN INDEX OR ON PREVIOUS DDCARD ***

This message is issued by U#UTIL1. If index entry information appears with this message, it indicates that the ddname on the ddcard being processed matches the ddname in the index entry. This form of the message occurs only if the run is not of the COMPILER or OVERLAY type and if the index entry with the matching ddname is of the type (System Group, User Group) which is being processed by the run. If no index entry information appears with this message, it indicates that the ddname on the ddcard being processed matches the ddname on a ddcard processed previously in the same run.

** NO ADD OR DEL FUNCTION SPECIFIED ***

This message is issued by UTILX; it indicates that no function control card (which must specify either ADD or DEL) has yet been encountered.

*** NOT ENOUGH FIELDS FOR FUNCTION SPECIFIED ***

This message is issued by UTILX. If the DEL function has been specified, this message indicates that this function could not be performed for the previous detail card because the first and fourth fields were not given. If the ADD function has been specified, this message appears if any of the six fields have been omitted.

*** NOT NEXT RELATIVE DSNUMBER ***

This message is issued by U#UTIL1 and may occur during a WORK/SWAP, SYSGROUP, or USRGROUP run. It indicates that the data set number (specified by the last two characters in the ddname on the ddcard being processed) is more than one greater than the highest data set number recorded in the index for that particular work/swap, system group, or user group (in the appropriate cluster). Since data sets can only be added in sequence with each group and cluster, UTILX should be used to obtain a listing of the index in order to determine which data set number may be added next.

*** RELATIVE DSNUMBER OVER IMPLEMENTATION LIMITS ***

This message is issued by U#UTIL1. During a WORK/SWAP run, this message indicates that the data set number (specified by the last two characters in the ddname on the ddcard being processed) is 20 or more. During a SYSGROUP or USRGROUP run, it indicates that the data set number is 80 or more. Only data set numbers of 0 to 19 are allowed for work/swap data sets, and system group and user group data sets may use only data set numbers 0 to 39 for the primary cluster and 40 to 79 for the alternate cluster.

*** THERE WAS AN ERROR IN THE PARM FIELD ON THE EXEC CARD. SEE THE CONSOLE LISTING FOR CORRECTION BY THE OPERATOR. ***

This message is issued by U#UTIL1; it indicates that the PARM field on the EXEC card was not correct; that is, was not COMPILER, SYSGROUP, USRGROUP, WORK/SWAP, or OVERLAY. An error message to this effect is also printed on the operator's console, and the operator is given a chance to enter a correct parameter.

*** UPDATE IMPOSSIBLE - INDEX FULL OR ENTRY NOT FOUND ***

This message is issued by UTILX. If the ADD function has been specified, this message indicates that this function could not be performed for the previous detail card because the index is full. If the DEL function has been specified, this message indicates that this function could not be performed for the previous detail card because there was no corresponding entry in the index.

*** VALIDATING - BAD I/O READING HOME ADDRESSES ***

This message is issued by U#UTIL1. During a COMPILER or WORKSWAP run, this message indicates that an I/O error condition occurred while attempting to read the home addresses on tracks to see if alternate tracks have been assigned.

*** VALIDATING - BAD OPEN ***

This message is issued by U#UTIL1. During a COMPILER or WORKSWAP run, this message indicates that OS/360 was unable to open (for input) the specified data set in order to check for flagged tracks and multiple extents.

*** VALIDATING - DATA SET HAS IMPROPER BOUNDARIES ***

This message is issued by U#UTIL1. On a COMPILER run, this message means that the data set (specified by the DD card being processed) crosses a cylinder boundary. On a WORKSWAP run, this message means that the data set does not begin and end at cylinder boundaries.

*** VALIDATING - FLAGGED TRKS FOUND ***

This message is issued by U#UTIL1. During a COMPILER or WORKSWAP run, this message indicates that bad tracks (for which alternate tracks have been assigned) have been found in the space allocated to the data set being processed. Since compiler and work/swap data sets cannot use alternate tracks, a different space, free from defective tracks, must be allocated for the data set.

*** VALIDATING - MULTIPLE EXTENTS FOUND ***

This message is issued by U#UTIL1. During a COMPILER or WORKSWAP run, this message indicates that the data set specified has had more than one extent allocated. Compiler and work/swap data sets cannot have multiple extents. To avoid this situation, the DD card used to create the data set should specify the CONTIG subparameter of the SPACE parameter. The RLSE subparameter is also recommended.

*** VALIDATING - MULTIPLE VOLUME DATA SET FOUND ***

This message is issued by U#UTIL1 and can occur on any type of run. It indicates that the data set specified does not reside on a single volume.

APPENDIX A: DEFINITION OF CODES FOR *STATUS COMMAND

USER'S STATUS CODES (STAT)

When *STATUS is issued, a hexadecimal code indicates the current status of the user. The codes and their associated commands are listed below:

Code	Associated Command	Code	Associated Command
01	RUN without parameters	1E	WIDTH
02	RUN with parameters	1F	CATALOG
03	LIST	20	ALLOW
04	LOAD	21	PROTECT
05	SAVE	22	POOL
06	RENUMBER, DELETE, EXTRACT, ADD	23	PULL
	REPLACE, MOVE, FIND, INSERT	24	*STATUS
07	MERGE, WEAVE	25	*USERS
08	TIME	26	*DATE
09	*CANCEL	27	Not used
0A	*VALIDATE	28	STATUS
0B	CLEAR	29	*REPORT
0C	*TELL	2A	KEY
0D	LOCK	2B	TAPE
0E	LOGON	2C	*BATCH
0F	OFF	2D	Not used
10	FILE	2E	STORE
11	PASSWORD	2F	CANCEL
12	PURGE	30	DSSTATUS
13	UNLOCK	31	JOBSTATUS
14	*DISABLE	32	Not used
15	*ENABLE	33	NOTIFY
16	*MESSAGE	34	SCAN
17	*WARN	35	SCRATCH
18	ECHO	36	SUBMIT
19	ENTER	37	*COBI
1A	HELP	38	PUNCH
1B	*OFF	39	SECURE
1C	NAME	3A	RELEASE
1D	*IGNORE		

TERMINAL CHANNEL STATUS CODES (TCHST)

When *STATUS is issued, a code indicates the status of the channel to which the terminal is attached. The codes and their meanings are listed below.

Code	Meaning
00	Idle (no command outstanding)
04	Writing text
08	Writing marks
0C	Reading text
10	Reading skip (ignore PCI)
14	Disable outstanding
18	Enable outstanding
1C	Prepare issued for break test
20	Disable outstanding (trouble) - message next
24	Issue disable next (normal)
28	Ignore interrupts (line out of service)
2C	Paper tape is halting
80	Recursive entry state to appendage--error recorded

TERMINAL FLAG BYTE CODES (TFLG1)

When *STATUS is issued, a code indicates the status of the terminal. The codes and their meanings are listed below.

Code	Meaning
80	Error has occurred in the enable sequence
40	Do not enable
20	Information message queued for this line
10	Line break has been received
08	Message routine is performing line folding
04	Program lines lost when inputting from paper tape
02	Warning message ready to print
01	Flag to get special sort error message

OS/360 IOS TERMINAL COMMUNICATIONS SWITCH CODES (IOSW)

When *STATUS is issued, a code indicates the status of the terminal IOS switch. The codes and their meanings are listed below.

Code	Meaning
00	Exit from start I/O appendage
04	Set up to issue halt I/O for break test
08	Switch for start I/O to post appendages for busy test
08	Last legal I/O vector. Must equal last legal value.

APPENDIX B: NONRESIDENT MODULE NUMBERS

When a module in the overlay detects an error which causes it to terminate, the module issues a message on the CALL-OS communications console. The message identifies the module by number, as follows:

Hexadecimal Number	Module Name
070	M#ACCT
080	M#STOR
090	M#CAT
0A0	M#CCDA
0B0	M#CCDI
0C0	M#CCME
0D0	M#CCOF
0E0	M#CCRE
0F0	M#CCST
100	M#CCTE
110	M#CCUS
120	M#CCVA
130	M#CCWA
170	M#DIR
1B0	M#ECHO
1C0	M#EDIT
200	M#HELP
230	M#LDRD
240	M#LIB
260	M#LIST
270	M#LOAD
280	M#LOG
2A0	M#MWSC
2B0	M#NAME
2F0	M#PASS
300	M#ESCN
310	M#RDSO
330	M#RUN
340	M#SAVE
350	M#SORT
380	M#STAT
3A0	M#TIME
3C0	M#WID
3D0	M#WRSO
550	M#CCBA
570	M#OBJR
590	M#CANCL
5B0	M#CBST
5C0	M#CCCO
5D0	M#IJCL
5E0	M#ISCAN
5F0	M#ISUB
600	M#JCL
610	M#NOTFY
620	M#SCAN
630	M#SCR
640	M#SUB
690	M#MREM
6A0	M#WEAV
6B0	M#SCAN

APPENDIX C: SAMPLE STARTUP DECK

This appendix contains a sample startup deck which could be used to initialize the CALL-OS system. The sample deck is followed by a summary of the initialization options available and the JCL statements in the deck. For detailed information on the startup deck, see the publication CALL-OS Executive and Utilities Program Description Manual.

SAMPLE DECK

The following is a sample startup deck for CALL-OS, containing the statements that could be present:

```
//CALLOS        JOB     MSGLEVEL=1,CLASS=B
//JOBLIB        DD      DSN=OSRTS.JOBLIB,DISP=SHR
//CALLINIT      EXEC    PGM=RTOS1,ROLL=(NO,NO),PARM='OPTION1,
//                      OPTION2,ETC'
//SYSABEND      DD      SYSOUT=A
//SYSPRINT      DD      SYSOUT=A
//INDEX         DD      DSN=OSRTS.INDEX,DISP=SHR
//RESMODS       DD      DSN=SYS1.PROCLIB(MEMBER),DISP=SHR
//CVLY          DD      DSN=OSRTS.OVLY,DISP=OLD
//BASIC         DD      DSN=OSRTS.BASIC,DISP=OLD
//FORTRAN       DD      DSN=OSRTS.FORTRAN,DISP=OLD
//PLI           DD      DSN=OSRTS.PLI,DISP=OLD
//PL2           DD      DSN=OSRTS.PL2,DISP=OLD
//SWAP00        DD      DSN=OSRTS.SWAP00,DISP=OLD
//SWAP01        DD      DSN=OSRTS.SWAP01,DISP=OLD
//SYSGRP00      DD      DSN=OSRTS.SYSGRP00,DISP=OLD
//SYSGRP01      DD      DSN=OSRTS.SYSGRP01,DISP=OLD
//AAABBB00      DD      DSN=OSRTS.AAABBB00,DISP=OLD
//AAABBB01      DD      DSN=OSRTS.AAABBB01,DISP=OLD
    .
    .
    .
//YYYZZZ00      DD      DSN=OSRTS.YYYZZZ00,DISP=OLD
//YYYZZZ01      DD      DSN=OSRTS.YYYZZZ01,DISP=OLD
//TWX           DD      UNIT=TERMGRP1
//T2741         DD      UNIT=TERMGRP2
//T2741E        DD      UNIT=TERMGRP3
//SYSJOBQ       DD      DSN=SYS1.SYSJOBQE,DISP=SHR
//CBNDX         DD      DSN=OSRTS.CBNDX,DISP=OLD
//CBJCL         DD      DSN=OSRTS.CBJCL,DISP=OLD
//CBSYSINA      DD      DSN=OSRTS.CBSYSINA,DISP=SHR
//CBSYSINB      DD      DSN=OSRTS.CBSYSINB,DISP=SHR
//SCAN01        DD      VOL=SER=COBI01,DISP=SHR,UNIT=2314
//SCAN02        DD      VOL=SER=COBI02,DISP=SHR,UNIT=2314
//SYSIN         DD      *
```

INITIALIZATION OPTION SUMMARY

When CALL-OS is initialized, certain options may be selected to tailor the system to the needs of the installation. These options appear in the parameter field of the EXEC statement (named CALLINIT in the sample) unless more than 100 characters of information is supplied; in this case, the options appear after the SYSIN DD statement.

The following is a list of the options in alphabetical order. Valid ranges and defaults are given where appropriate; if the operator can override the option or its default, the specific operator command is indicated.

Option Explanation

ACTIME=nnn

Specifies the number of minutes that is to elapse between accounting checkpoints. The default is 30 minutes.

ANYJNAME

Specifies that user-supplied job names are to be used for all COBI jobs. If COBI is not used, this option is ignored if present; if COBI is used and the option is omitted, COBI assigns a name to every job submitted.

AUTRDR

Specifies that COBI is to start a reader for the COBI input data sets when necessary. If COBI is not used, this option is ignored if present; if COBI is used and the option is omitted, the operator must start a reader when instructed by COBI. (See "Starting a Reader" in the chapter "Online Operations".) The operator may override the mode of starting a reader with the *COBI RESET, AUTRDR command as described in the chapter "Operator Command Language".

CBCLASS=z

Specifies the output class to be used for COBI and must be a valid output class (0 through 9 or A through Z). If CALL-OS is not executing and COBI jobs must still be processed, the operator starts DIBWTR, which processes this class. (See "Starting a Writer" in the Introduction). If COBI is not used, this option is ignored if present; if COBI is used and the option is omitted, the default is output class Z.

If the operator displays the job queue, COBI jobs appear in this class. Information in this class is switched to another class (specified by the OSCLASS parameter) for processing by an OS/360 output writer. If the operator attempts to change the COBI class to the OS/360 class or starts an OS/360 writer for the COBI class, the output will be lost.

COMCON=nnn

Specifies the logical line number of the communications console for CALL-OS. The default is logical line number 2. If COMCON=0 is specified, all messages intended for the communications console appear on the OS/360 system operator's console preceded by the identifier DIBEX001.

COMTSL=(nnn,name)

Specifies the time slice in seconds to be allotted to the indicated compiler or compiler phase. The minimum is one second and the maximum is ten seconds. The default is one second for BASIC, and three seconds each for FORTRAN, PLI, and PL2.

Note: These times are for the IBM System/360, Model 50.

Option	Explanation
DFLINK=nnn	Specifies the maximum number of half tracks to be allocated to each data file; the number must be within the range 4 through 100. The default is 100 half tracks.
DSPACE=(ppp,sss)	Specifies the primary and secondary allocation in tracks for the scannable output data sets created by COBI jobs. If COBI is not used, this option is ignored if present; if COBI is used and the option is omitted or a zero value is specified, the default is ten tracks for the primary and ten for the secondary.
IPBUFS=nnn	Specifies the number of 24-byte buffers (pots) to be allocated per line. The default is four pots per line with a minimum total of 60 pots in the system. The maximum is 15 pots per line, or the default total of 60 if the number of lines is less than or equal to four.
LCSRES=aaaaaaaa	Specifies which portions of CALL-OS system are to be loaded into Hierarchy 1 storage. Up to nine different portions may be indicated by letter; the allowable letters are: B - 256-byte buffers C - compiler area D - data control blocks (DCB) G - COBI tables and bit strings J - new and old job areas O - overlay buffer P - pots (24-byte buffers) and terminal translate tables S - sort buffer U - user terminal tables (UTT)
MAXDCB=nn	Specifies the maximum number of users permitted to scan data sets at the same time when COBI is used. If COBI is not used, this option is ignored if present; if COBI is used and the option is omitted, the default is one user for every ten lines with a minimum of two users.
NOCOBI	Specifies that COBI is not to be active for this session of CALL-OS.
NOSORT	Specifies that the sort buffer is not to be present in the system.
NOTRACE	Specifies that the trace table, if one is present in the OS/360 system, is not to contain entries for CALL-OS.

Option Explanation

OPBUFS=nnn

Specifies the total number of 256-byte buffers to be present in the system. The default is one buffer per three lines with a minimum of five buffers. The maximum is one buffer per line or the default total of five if the number of lines is less than or equal to five.

OSCLASS=a

Specifies the output class to be used for COBI job output after processing by COBI; it must be a valid output class (0 through 9 or A through Z). It must not be the same output class specified in the CBCLASS parameter. If COBI is not used, this option is ignored if present; if COBI is used and the option is omitted, the default is output class A.

RDRQTY=nnn

Specifies the maximum number of jobs to be accepted into a COBI input data set before the data sets are switched. The number specified must be within the range 0 through 999; a value of 0 makes the function inoperative. If COBI is not used, this option is ignored if present; if COBI is used and this option is omitted, the default is ten jobs. The operator may override the number of jobs with the *COBI RESET,RDRQTY command; the operator may cause the number to be ignored with the *COBI ANYBATCH command. The *COBI command is described in the section "Operator Command Language".

RDRTIM=mmm

Specifies the number of minutes to elapse before the COBI input data sets are switched. The number specified must be within the range 0 through 999; a value of 0 makes the function inoperative. If COBI is not being used, this option is ignored if present; if COBI is being used and the option is omitted, the default is 30 minutes. The operator may override the number of minutes with the *COBI RESET,RDRTITM command; the operator may cause the number to be ignored with the *COBI ANYBATCH command. The *COBI command is described in the section "Operator Command Language".

RUNTSL=(nnn,mmm)

Specifies the time slice in seconds to be allocated to the new and old jobs; the time slice must be within the range 0.5 through 5.0 seconds for new jobs and 1 through 20 seconds for old jobs. The default is three seconds for new jobs and ten seconds for old jobs.

Note: These times are the times recommended for the IBM System/360, Model 50.

SCANDS=(index1,index2)

Specifies one or two high-level index qualifiers of the system data sets which the COBI user is permitted to scan. If COBI is not used, this option is ignored if present; if COBI is used,

Option	Explanation
	and the option is omitted, the user is not permitted to scan any data set which does not contain his user number as part of the data set name.
SHRTSL=ttt	Specifies the time slice increment of time employed in sharing time with background processing. The time slice must be less than that specified for old jobs (see RUNTSL). The maximum is ten seconds, the minimum is 0.1 seconds, and the default is one second. The operator may control the percentage and mode of allocation with the *BATCH command, as described in the section "Operator Command Language".
SYSCON=(ppp,aaa)	Specifies the logical line number of the primary and alternate command consoles. The default is logical line number 1 for the primary console and logical line number 0 for the alternate.
UNITNM=name	Specifies a group of devices which are to contain the volumes to be used for scannable output data sets created by COBI jobs. It may specify a device type or a generic name; if a generic name is used, the name must have been specified in the UNITNAME macro instruction during system generation. In addition, a device address may be specified. If COBI is not used, this option is ignored if present; if COBI is used and the option is omitted, the default is 2314.

JCL STATEMENT SUMMARY

The following text summarizes the JOB, EXEC, and DD statements required to initialize CALL-OS.

JOB Statement

The JOB statement in the startup deck for CALL-OS must meet certain requirements; it:

- must specify a job name (in the sample the name is CALLOS); this name appears in the termination message issued by OS/360 when CALL-OS terminates abnormally,

- must specify ROLL=(NO,NO) unless this parameter is specified on the EXEC statement,

- should assign a unique job class to CALL-OS, and

- must specify a REGION for MVT unless this parameter is specified on the EXEC statement.

EXEC Statement

The EXEC statement in the startup deck for CALL-OS must meet certain requirements; it:

- must specify the CALL-OS program name; that is PGM=RTOS1,
- must specify ROLL=(NO,NO) unless this option is specified on the JOB statement,
- may specify initialization options in the PARM field if less than 100 characters of information is supplied, and
- must specify a REGION for MVT unless this parameter is specified on the JOB statement.

DD Statements

The data set names used in the startup deck may vary with the installation; they must be the names supplied for the CALL-OS system build utility program. The names used in the sample deck are those created and cataloged by default during system build. The names of DD statements must have the formats shown in the sample deck.

The following is an alphabetical list of the DD statements by name that may appear in a startup deck for CALL-OS:

DD STATEMENT	Explanation
aaabbbnn yyyzzznn	A DD statement with a name of six letters followed by two numbers defines a group of users to be permitted access to the system for this session. At least one DD statement of this form is required. The letters may be any alphabetic character from A through Z and indicate the range of users in the group: the first three letters specify the low order user and the last three letters specify the high order user. (For example, the AAABBB00 DD statement specifies that users with numbers in the range from AAA001 through BBB999 may use CALL-OS.) The numbers indicate the cluster to which the data sets belong, 00 through 39 indicate the primary cluster and 40 through 79 indicate the alternate cluster. If only one DD statement is required for a group, the number must be either 00 or 40. If more than one DD statement is required for a group (the letters in the DD names are identical), the numbers must start with 00 (or 40) and must be in sequence through 39 (or 79).
BASIC	Specifies the location of the BASIC compiler and must be omitted if BASIC is not supported.
CBNDX CBJCL CBSYSINA CBSYSINB	Define the COBI index, the JCL data set, and the two input data sets, respectively. If COBI is to be used, all four statements must be present; if COBI is not used, these statements are ignored if present.
FORTRAN	Specifies the location of the FORTRAN compiler and must be omitted if FORTRAN is not supported.

DD Statement — Explanation

INDEX — Defines the CALL-OS index; this statement is required.

JOBLIB — Defines the private library which contains the CALL-OS modules. Either this statement or a STEPLIB DD statement must be present unless the CALL-OS modules reside in SYS1.LINKLIB.

OVLY — Defines the data set to contain the overlay modules. This statement must be present if the system is not totally resident (see RESMODS DD statement).

PLI
PL2 — Define the location of the first and second phases, respectively, of the PL/I compiler. If PL/I is used, both statements must be present; if PL/I is not supported, both statements must be omitted.

RESMODS — Defines the module residency list to be used. The list to be used is indicated by the member name. The allowable IBM-supplied names are:

RTOSALL -	totally resident system
RTOSNONE -	only the nucleus is to be resident
RTOSLLRS -	modules for load, list, save, and run functions are to be resident
RTOSUSER -	all modules resident except those associated with operator commands

The first case is the only one in which the OVLY DD statement is not required; for all the other cases, the OVLY DD statement must be present.

SCAN01
SCAN02 — A DD statement with a name of SCAN followed by two characters defines a volume which contains scannable data sets for COBI. If COBI is used and output data sets are to be scanned; at least one such DD statement is required. If the SCANDS option is specified, one DD statement must be provided for each volume containing scannable system data sets. If COBI is not used, these DD statements should be omitted.

STEPLIB — Defines the private library which contains the CALL-OS modules. Either this statement or a JOBLIB DD statement must be present unless the CALL-OS modules reside in SYS1.LINKLIB.

SWAP00
SWAP01 — A DD statement with a name of SWAP followed by two numbers defines a work/swap data set. At least one DD statement of this form must be

DD Statement	Explanation
	supplied. If more than one is present, the numbers must be in order from 00 through 19.
SYSABEND	Defines the output class for dumps if CALL-OS terminates abnormally.
SYSGRP00 SYSGRP01	A DD statement with a name of SYSGRP followed by two numbers defines a system data set. The numbers must be in the range 00 through 39 for the primary cluster and 40 through 79 for the alternate cluster. At least one DD statement of this form must be supplied. If more than one is present, they must be from only one cluster and in order from 00 (or 40).
SYSIN	Defines the initialization option data set either if 100 characters of information are entered in the PARM field of the EXEC statement and more information is required, or if the PARM field on the EXEC statement is not specified; no more than 400 characters of information may be specified altogether.
SYSJOBQ	Defines the system job queue data set when COBI is used. If COBI is not used, this statement is ignored if present.
SYSPRINT	Defines the output class for output from the *REPORT command and the *COBI-P status commands.
TWX T2741 T2741E	Define the terminals to be supported by the system: teletype, 2741 correspondence, and 2741 EBCD, respectively. This is the sequence in which logical line numbers are assigned; however, the order of the DD statements has no effect on the assignment of line numbers. At least one of these DD statements must be present.

GH20-0788-2

CALL—OS V 2 OM Printed in U.S.A. GH20-0788-2

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

READER'S COMMENT FORM

CALL—OS Version 2
Operator's Manual

GH20-0788-2

Please comment on the usefulness and readability of this publication, suggest additions and deletions, and list specific errors and omissions (give page numbers). All comments and suggestions become the property of IBM. If you wish a reply, be sure to include your name and address.

COMMENTS

fold

fold

fold

fold

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.
FOLD ON TWO LINES, STAPLE AND MAIL.

GH20-0788-2

YOUR COMMENTS PLEASE...

Your comments on the other side of this form will help us improve future editions of this publication. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material.

Please note that requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or the IBM branch office serving your locality.

fold fold

FIRST CLASS
PERMIT NO. 1359
WHITE PLAINS, N.Y.

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY...

IBM Corporation
1133 Westchester Avenue
White Plains, N.Y. 10604

Attention: Technical Publications

fold fold

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
[U.S.A. only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

READER'S COMMENT FORM

CALL–OS Version 2
Operator's Manual

GH20-0788-2

Please comment on the usefulness and readability of this publication, suggest additions and deletions, and list specific errors and omissions (give page numbers). All comments and suggestions become the property of IBM. If you wish a reply, be sure to include your name and address.

COMMENTS

fold

fold

fold

fold

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A. FOLD ON TWO LINES, STAPLE AND MAIL.

GH20-0788-2

YOUR COMMENTS PLEASE...

Your comments on the other side of this form will help us improve future editions of this publication. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material.

Please note that requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or the IBM branch office serving your locality.

fold fold

FIRST CLASS
PERMIT NO. 1359
WHITE PLAINS, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY...

IBM Corporation
1133 Westchester Avenue
White Plains, N.Y. 10604

Attention: Technical Publications

fold fold

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
[U.S.A. only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

IBM / Technical Newsletter

Base Publ. No. GH20-0788-2

This Newsletter No. GN20-2781

Date July 31, 1972

Previous Newsletter Nos. None

CALL-OS
Operator's Manual
Program Number 360A-CX-42X

This Technical Newsletter is intended only for those who wish to make the base publication apply to Version 1, Modification Level 2. It provides replacement pages for the subject manual. These replacement pages remain in effect for subsequent versions and modifications unless specifically altered. Pages to be inserted and/or removed are listed below:

3–4
11–14
21–24
55–56
65–68
81–84
84.1–84.2 added
89–90
95–96
99–100
109–110
117–118
125–126
131–134
139–140

A vertical rule in the left margin indicates a change. Absence of a vertical rule on a page bearing a "revised" notice means only that existing copy has been moved or that a minor typographical error has been corrected.

Please file this cover letter at the back of the manual to provide a record of changes.

Note: The IBM Operating System is commonly referred to as OS. For consistency, the term OS/360 has been replaced by the term OS on all changed pages of this manual. Users of the manual should regard OS and OS/360 as synonymous.

IBM Corporation, Technical Publications Dept., 1133 Westchester Avenue, White Plains, N. Y. 10604

Printed in U.S.A.