

60493300

CONTROL DATA
CORPORATION

CYBER COMMON UTILITIES REFERENCE MANUAL

**CONTROL DATA®
CYBER 170 SERIES
CYBER 70 SERIES
7600 SERIES
6000 SERIES
COMPUTER SYSTEMS**

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

| Feature | Page | Revision |
|---------|--------------|----------|
| | Cover | — |
| | Title Page | — |
| | ii thru vii | A |
| | 1-1 thru 1-3 | A |
| | 2-1 thru 2-6 | A |
| | 3-1 thru 3-7 | A |
| | A-1, A-2 | A |
| | B-1, B-2 | A |
| | Cmt Sheet | A |
| | Return Env | — |
| | Back Cover | — |

| Feature | Page | Revision |
|---------|------|----------|
| | | |



PREFACE

This publication describes the utilities that operate under the following operating systems:

KRONOS 2.1 for the CONTROL DATA®
CYBER 70/Model 72, 73, 74, and 6000 Series
Computer Systems.

NOS 1.0 for the CONTROL DATA CYBER
170, CYBER 70/Model 72, 73, 74 and 6000
Series Computer Systems.

SCOPE 2.1 for the CONTROL DATA CYBER
70/Model 76 and 7600 Computer Systems.

SCOPE 3.4 for the CONTROL DATA CYBER
70/Model 72, 73, 74, and 6000 Series Com-
puter Systems.

The utilities described are:

COPYL/ Maintains a single binary file.
COPYLM (Replaces the COPYL utility under
SCOPE 3.4 and COPYL and
COPYLM utilities under SCOPE 2.1.)

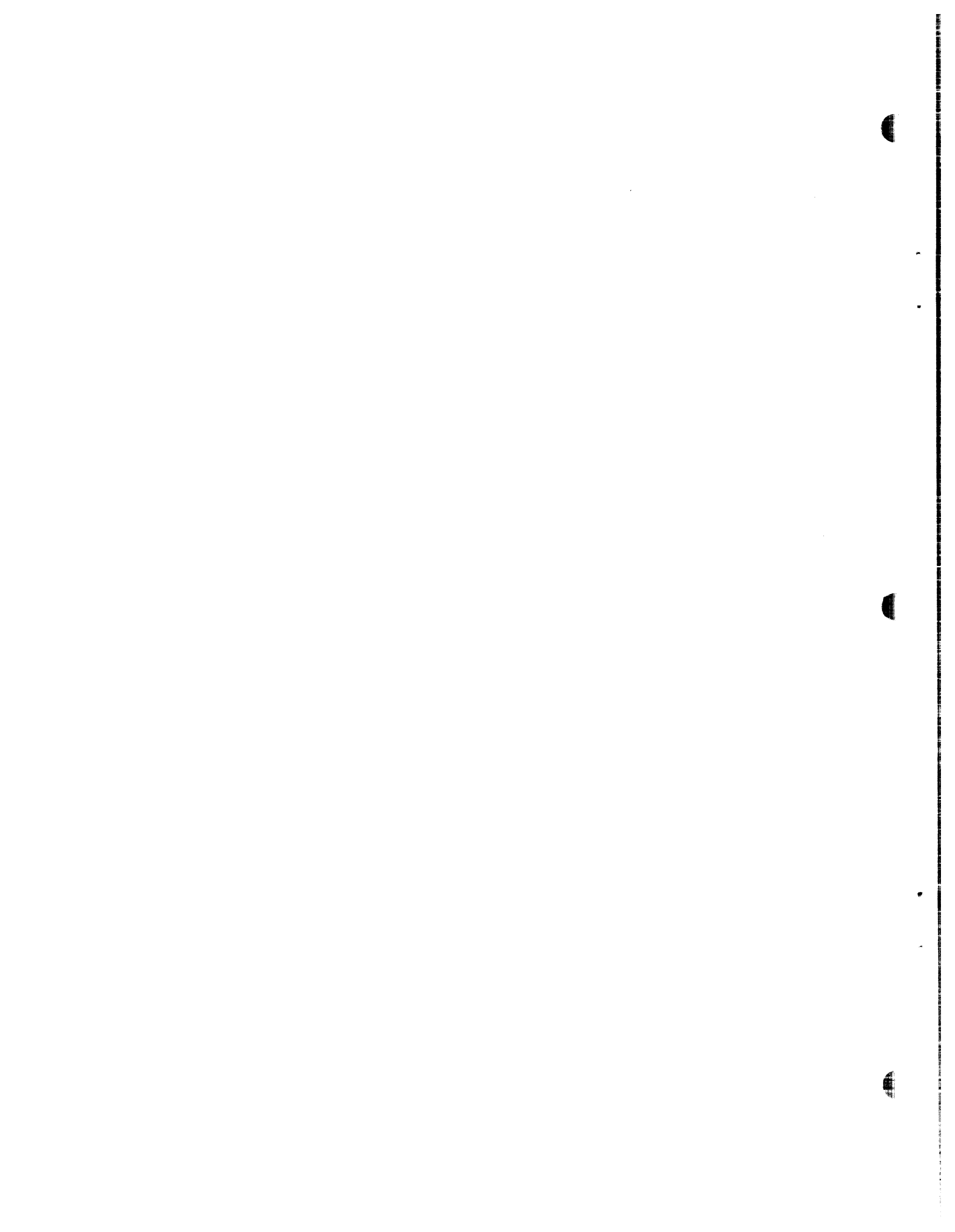
ITEMIZE Lists information about the contents
of a binary file or multifile.

Other utility programs for the above operating
systems are documented in their respective operating
system manuals.

Other documents of interest:

| <u>Publication</u> | <u>Publication Number</u> |
|-----------------------------|---------------------------|
| KRONOS 2.1 Reference Manual | 60407000 |
| NOS 1.0 Reference Manual | 60435400 |
| SCOPE 3.4 Reference Manual | 60307200 |
| SCOPE 2.1 Reference Manual | 60342600 |

This product is intended for use only as
described in this document. Control Data
cannot be responsible for the proper func-
tioning of undescribed features or undefined
parameters.



CONTENTS

| | | | | | |
|---|----------------------------|-----|---|--|-----|
| 1 | INTRODUCTION | 1-1 | 2 | COPYL/COPYLM: BINARY COPY WITH REPLACEMENT | 2-1 |
| | File Structure Definitions | 1-1 | 3 | ITEMIZE: LIST CONTENTS OF BINARY FILE | 3-1 |
| | Types and Names of Records | 1-1 | | | |

APPENDIXES

| | | | | | |
|---|-------------------------|-----|---|---------------------|-----|
| A | STANDARD CHARACTER SETS | A-1 | B | DIAGNOSTIC MESSAGES | B-1 |
|---|-------------------------|-----|---|---------------------|-----|

FIGURES

| | | | | | |
|-----|---|-----|-----|---|-----|
| 2-1 | COPYL and COPYLM Dayfile Example Using R and A Parameters | 2-3 | 3-1 | ITEMIZE Example of Records on KRONOS Deadstart Tape | 3-4 |
| 2-2 | Contents of Files from Figure 2-1 | 2-4 | 3-2 | ITEMIZE Example Using U and E Parameters | 3-5 |
| 2-3 | COPYL Example Showing Use of T Parameter | 2-5 | 3-3 | ITEMIZE Example of SCOPE 3.4 Deadstart Tape | 3-6 |
| 2-4 | COPYL Example Showing COS Type Record Handling | 2-6 | 3-4 | ITEMIZE Example of Sequential UPDATE File Records and Dense Listing | 3-7 |

TABLES

| | | | | | |
|-----|---|-----|-----|------------------------------------|-----|
| 1-1 | Determining Types of Records | 1-3 | 3-1 | Types of Records Listed by ITEMIZE | 3-1 |
| 2-1 | Types of Records Replaced by COPYL and COPYLM | 2-1 | A-1 | Standard Character Sets | A-2 |
| | | | B-1 | Diagnostic Messages | B-2 |

The terminology used for files and records differs somewhat within the different environments of the various operating systems. Generally, terms used with KRONOS, NOS, and SCOPE 3.4 are similar; terms used with SCOPE 2.1 are different, reflecting SCOPE 2.1 use of Record Manager within the operating system.

The physical representation of files and records differs among operating systems and among different devices within the same operating system. The logical interpretation of physical entities varies less. The CYBER utilities recognize logical structures applicable to all the operating systems.

Terms applicable to this manual and the common CYBER utilities are discussed below. For a complete discussion of possible file formats, consult the reference manuals for the respective operating systems.

FILE STRUCTURE DEFINITIONS

A file is that collection of information known to an operating system by a logical file name. A file begins at beginning-of-information and ends at end-of-information. No user information exists outside these boundaries.

A division smaller than beginning-of-information to end-of-information is known by several different names, depending on the operating system.

Under SCOPE 2.1, information between beginning-of-information and end-of-information can be divided into partitions which, in turn, can be divided into sections. The term record is restricted to a data grouping in a format specified to Record Manager by an RT mnemonic parameter in a file information table in memory or by a FILE control statement. Examples of Record Manager record types are W, in which a control word provides record length, and F, which is a fixed length record.

Under KRONOS, NOS and SCOPE 3.4, information between beginning-of-information and end-of-

information is divided into logical records terminated by a marker containing a level number. A marker containing a level 17 establishes the next smaller division between beginning- and end-of-information; the level 17 marker is commonly termed an end-of-file marker. A record terminated with a level 0 marker is smaller than a level 17 record; the level 0 marker is commonly termed an end-of-record marker. The physical file formats that produce the logical end-of-file and end-of-record are the formats internal to the operating system (KRONOS and NOS I (internal), SI (SCOPE internal), or X (external) tape format or mass storage; SCOPE 3.4 mass storage and tape SCOPE logical record format).

Terms logically equivalent among operating systems are:

| <u>SCOPE 2.1</u> | <u>KRONOS, NOS, SCOPE 3.4</u> |
|------------------|---|
| file | file |
| partition | level 17 record; also referred to as a file |
| section | level 0 record; also referred to as a record |

In this manual, terms from KRONOS, NOS and SCOPE 3.4 are used predominantly. As a result, the SCOPE 2.1 end-of-partition becomes an end-of-file for the purposes of many discussions, and the SCOPE 2.1 section is referred to as a record.

TYPES AND NAMES OF RECORDS

The type and the name of a record are determined by the COPYL and ITEMIZE utilities from information contained within the record. If the record begins with a prefix table, the record name is obtained from that table and the type of the record is determined from the first word following the prefix table. If the first word in the record is not a prefix table, but is a recognizable format, the format determines type. Any record that has neither a prefix table nor a recognizable format is classed as a DATA type record.

PREFIX TABLE USE

Prefix tables exist, unless they have been specifically suppressed, for:

Programs assembled or compiled under any operating system

System text overlays

KRONOS and NOS user library header records and directory records

MODIFY program library decks, common decks, or directory records

The prefix table is the first of the ordered set of binary tables that form object programs. The tables consist of a header word with an octal table type identifier followed by varying amounts of control information that instruct system routines such as the loader or that contain the program code.

The prefix table is identified by octal digits 7700 in bits 48-59 of its first word; consequently, it is often referred to as a 77 or 7700 table. Information in the prefix table, which originates with the assembler or other system routine that creates the table, specifies items such as the date created and the system on which the job was executed.

Although some of the records may contain display coded data (loader directives, for instance, are coded), they are considered binary records.

OTHER RECORD IDENTIFIERS

If a prefix table is not present, the first word in a record is examined in a search for a recognizable format. An UPDATE sequential program library, for example, is identified by the characters CHECK in the word.

If a record meets the criteria for a given type of record, the utilities identify it as such. For instance, a load file beginning with a job card may be identified as type COS, TEXT, or DATA, depending on the particular characters in the job card.

Table 1-1 summarizes types of records and the criteria used to determine them.

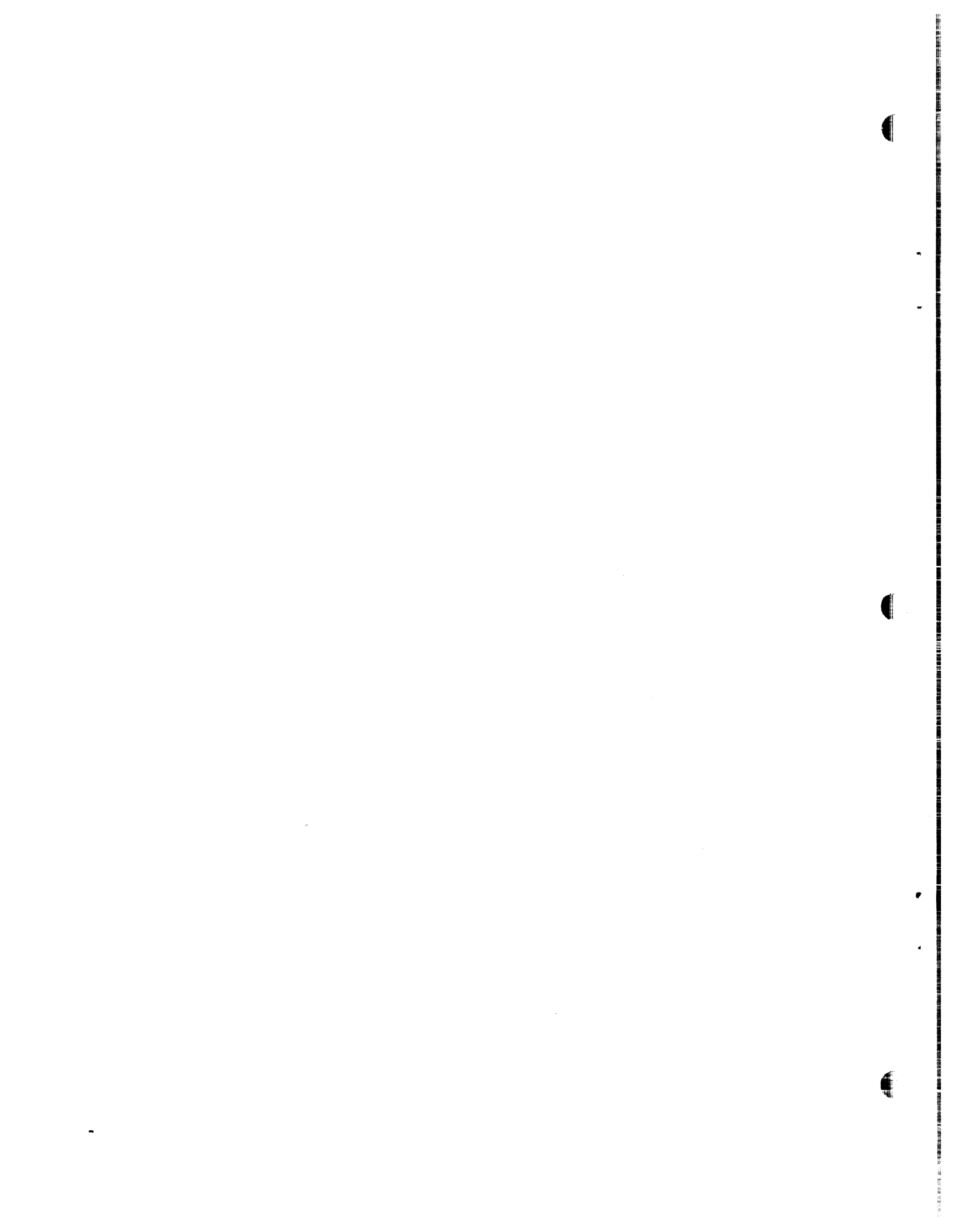
RECORD NAMES

If a record begins with a prefix table, bits 18-59 of the second word of the table determine the record name. If a record does not begin with a prefix table, bits 18-59 of the first word of the record are used as the record name.

Records typed as DATA, ACF, UCF, and UPL do not have names.

TABLE 1-1. DETERMINING TYPES OF RECORDS

| Type of Record | Record Description | Type Determined By |
|----------------|--|--|
| ABS | Central processor overlay with one or more named entry points | 51 table; or 53 table with bit 17=1 |
| ACF | MODIFY compressed compile file | Second word of 77 table has non-zero in bits 00-17 |
| COS | Chippewa format central processor program; COMPASS or FORTRAN program with errors that suppressed binaries | No 77 table and first word of record has bit 17=0, bit 59=0, and bits 0-16 non-zero |
| DATA | Not any other described record type | Unrecognizable by criteria defined in these tables |
| LIBNT | Library name table record | SCOPE 3.4 deadstart tape position |
| OPL | MODIFY program library deck | 7001 table with 0 word count |
| OPLC | MODIFY program library common deck | 7002 table with 0 word count |
| OPLD | MODIFY program library directory | 7000 table with 0 word count |
| OVL | Central processor overlay with one unnamed entry point (no ENTRY statement in program); system text | 50 table; or 53 table with bit 17=0 |
| PPNT | Peripheral processor program name table | SCOPE 3.4 deadstart tape position |
| REL | Relocatable central processor program | 34 table |
| SDR | Special deadstart record | SCOPE 3.4 deadstart tape position |
| TEXT | Text record such as a KRONOS procedure file | No 77 table and first word has all zeros in bits 0-17 |
| UCF | UPDATE compressed compile file | 77 table with 0 word count |
| ULIB | KRONOS user library header record | 76 table |
| UPLx | UPDATE sequential program library with x master control character | No 77 table and characters CHECK in bits 30-59; control character obtained from bits 0-5 |
| 6PP | 6000 Series peripheral processor overlay | 77 table with three-character name in header word |
| 7PP | 7000 Series peripheral processor overlay | 52 table |



COPYL copies an old master file to a new master file, substituting records from a replacement file for matching records on the old master, and optionally adding records to the end of the new master file. Records are considered to match if they have the same type and name, or optionally, the same name. COPYL operates with binary or text records. It is commonly used to maintain files of procedures or subroutines.

COPYL and COPYLM differ only in the handling of multiple occurrences of a record on the old master. COPYL uses each record on the replacement file only once, replacing the first matching record from the old master file. COPYLM uses the first matching record encountered on the replacement file to replace each matching record from the old master file. COPYL can be used for multiple replacement only if multiple copies of the record are on the replacement file.

The master and replacement files must reside on mass storage or a binary tape in a format internal to the operating system (SCOPE 3.4 format tape for SCOPE 3.4; I, SI, or X format for KRONOS and NOS; record type S or W for SCOPE 2.1). Only a single file terminated by a level 17 record marker (SCOPE 2.1 partition) is processed by a single call to COPYL or COPYLM.

Order of records on the replacement file is not significant. Records on the new master file are in the same order as they were on the old master file.

COPYL issues dayfile messages (appendix B) during processing; no other printed output is produced.

A field length of 12000 octal is required for execution.

COPYL replaces only the types of records listed in table 2-1. Any record on the old master file that is not recognized as one of the listed types is copied to the new master file without further processing.

TABLE 2-1. TYPES[†] OF RECORDS REPLACED BY COPYL AND COPYLM

| Type | Description |
|------|--|
| ABS | Central processor overlay with one or more named entry points |
| COS | Chippewa format central processor program; FTN binary with compilation errors or COMPASS binary with assembly errors |
| OPL | MODIFY old program library deck |
| OPLC | MODIFY old program library common deck |
| OVL | Central processor overlay with one unnamed entry point including system texts |
| REL | Relocatable central processor program |
| TEXT | Text record (e.g. KRONOS procedure) |
| 6PP | 6000 Series peripheral processor program |
| 7PP | 7000 Series peripheral processor program |

[†]For additional information about how these types are determined, consult table 1-1.

Control statement format:

COPYL(oldlfn, replfn, newlfn, last, flag) Single replacement

COPYLM(oldlfn, replfn, newlfn, last, flag) Multiple replacement

All parameters are optional and position dependent:

- oldlfn Logical file name of the old master file; default name is OLD.
- replfn Logical file name of the replacement file; default name is LGO.
- newlfn Logical file name of the updated master file; default name is NEW.
- last Name of the last record on oldlfn to be processed. If last is not specified, all records on oldlfn are processed.
- flag Processing options; default is do not select option.
 - R Rewind oldlfn and newlfn before processing. (Replfn is always rewound before and after processing.)
 - A Append to the end of newlfn all replfn records that do not match any on the oldlfn.
 - T Omit check for matching type of record.

These options can be specified by combining one or more letters in any order, as TRA, AR, RT, or TR.

The R option affects master file positioning only before processing. If R is specified, both the old and new master files are rewound to beginning-of-information before processing. In the absence of R, the user is responsible for positioning these files. After processing, the old master file is positioned after the end-of-file (end-of-partition) that stopped processing. COPYL writes an end-of-file and does not further position the new master file. The R option does not affect the file of replacement records, since the replacement file always is rewound before and after processing.

The A option causes unmatched records from the replacement file to be copied to the new master file following the last record copied from the old master file. If A is not selected, records on the replacement file are ignored and an informative message is issued when they do not match records on the existing file.

The T option determines whether the record name, or name and type, will be used to identify a matching record. If T is not selected, COPYL considers records to match only if both the type and name are the same. If T is selected, however, COPYL considers records to match if they have the same name, regardless of type. Otherwise, any replacement file record that is not of a type listed in table 2-1 is ignored without comment.

Records of type COS (such as result from a compilation or assembly of programs containing errors) are handled in a special manner. If a COS type record appears on a replacement file, it is always ignored without comment. If a COS type record appears on the old master file and a record with the same name but a type other than COS appears on the replacement file, the replacement record with the new type is always copied to the new master file. The match on type is suppressed for COS type records from the old master file even when the T option is not specified. A COS type record on the old master file that is not replaced is copied to the new master file without change.

Figure 2-1 shows the dayfile from a job that produces four new files (NEW, NEW1, NEW2, and NEW3), using the same old master file (OLD) and the same replacement file (LGO). Figure 2-2 identifies the contents of OLD and LGO (figure 2-2a-b) and the newly created files using the ITEMIZE utility.

Figure 2-3 shows an example of the T parameter of COPYL. The old master file (figure 2-3b) contains two absolute records named COPYL and ITEMIZE, as shown by the execution of the ITEMIZE utility. A new relocatable record with the name COPYL is created on replacement file XXX (figure 2-3c). Executing the COPYL utility without the T parameter produces a new master file, NEW1, that is identical to OLD (figure 2-3d). When the T parameter is used, records are replaced if the name matches even though the type may differ, so that NEW2 (figure 2-3e) is not the same as OLD.

Figure 2-4 shows that the T parameter is not required to suppress record type checking when a COS type record on the old master file is to be replaced. A COS type record is presumed to be a record with compilation or assembly errors when a record with a matching name appears on the replacement file. A COS type record on the replacement file does not suppress the type check, however, and in the absence of the T parameter a COS type record cannot be written to the new master file.

```

ATTACH OLD,OLDMASTERFILE2,ID=M0.
PF CYCLE NO. = 001
ITEMIZE(OLD)
ITEMIZE COMPLETE.
MAP,OFF.
FTN.
      .126 CP SECONDS COMPILATION TIME
ITEMIZE(LGO)
ITEMIZE COMPLETE.
REWIND(OLD,LGO) ← Rewind files.
COPYL. ← Default file names assumed.
  UPDATED -- REL / SURA      Only records existing on OLD are
  UPDATED -- REL / SUBB      copied to NEW; SUBZ is not on OLD.
COPYL DID NOT FIND -- REL / SUBZ
COPYL COMPLETE.      Only first record named SUBB is replaced.
ITEMIZE(NEW)
ITEMIZE COMPLETE.
COPYL(OLD,LGO,NEW1,,RA) ← R rewinds OLD,NEW1.
  UPDATED -- REL / SURA      A causes records on LGO to be
  UPDATED -- REL / SUBB      added to NEW1 if they did not replace
APPENDED -- REL / SUBZ      records on OLD.
COPYL COMPLETE.
ITEMIZE(NEW1)
ITEMIZE COMPLETE.
COPYLM(OLD,LGO,NEW2,SUBD,P) ← SUBD will be last record processed on
  UPDATED -- REL / SURA      OLD, effectively deleting any records after
  UPDATED -- REL / SUBB      SUBD, including third SUBB record.
  UPDATED -- REL / SUBB      COPYLM replaces two records named SUBB.
COPYL DID NOT FIND -- REL / SUBZ
COPYL COMPLETE.
ITEMIZE(NEW2)
ITEMIZE COMPLETE.
COPYLM(,,NEW3,,R) ← COPYLM replaces all occurrences of record
  UPDATED -- REL / SURA      SUBB.
  UPDATED -- REL / SUBB
  UPDATED -- REL / SUBB
  UPDATED -- REL / SUBB
COPYL DID NOT FIND -- REL / SUBZ
COPYL COMPLETE.
ITEMIZE(NEW3)
ITEMIZE COMPLETE.

```

Figure 2-1. COPYL and COPYLM Dayfile Example Using R and A Parameters

a. Contents of Old Master File OLD.

| REC | ITEMIZE OF OLD NAME | TYPE | FILE LENGTH | 1 CKSUM |
|-----|---------------------|------|-------------|---------|
| 1 | SUBA | REL | 50 | 3703 |
| 2 | SUBB | REL | 34 | 0727 |
| 3 | SUBC | REL | 37 | 5657 |
| 4 | SUBB | REL | 34 | 0727 |
| 5 | SUBD | REL | 40 | 6320 |
| 6 | SUBB | REL | 34 | 0727 |

* EOF * SUM = 273

c. Contents of NEW Produced by: COPYL.

| REC | ITEMIZE OF NEW NAME | TYPE | FILE LENGTH | 1 CKSUM |
|-----|---------------------|------|-------------|---------|
| 1 | SUBA | REL | 32 | 1435 |
| 2 | SUBB | REL | 40 | 2643 |
| 3 | SUBC | REL | 37 | 5657 |
| 4 | SUBB | REL | 34 | 0727 |
| 5 | SUBD | REL | 40 | 6320 |
| 6 | SUBB | REL | 34 | 0727 |

7 * EOF * SUM = 261

e. Contents of NEW2 produced by: COPYLM(OLD,LGO,NEW2,SUBD,R)

| REC | ITEMIZE OF NEW2 NAME | TYPE | FILE LENGTH | 1 CKSUM |
|-----|----------------------|------|-------------|---------|
| 1 | SUBA | REL | 32 | 1435 |
| 2 | SUBB | REL | 40 | 2643 |
| 3 | SUBC | REL | 37 | 5657 |
| 4 | SUBB | REL | 40 | 2643 |
| 5 | SUBD | REL | 40 | 6320 |
| 6 | * EOF * | | SUM = | 231 |

b. Contents of Replacement File LGO.

| REC | ITEMIZE OF LGO NAME | TYPE | FILE LENGTH | 1 CKSUM |
|-----|---------------------|------|-------------|---------|
| 1 | SUBZ | REL | 35 | 2312 |
| 2 | SUBB | REL | 40 | 2643 |
| 3 | SUBA | REL | 32 | 1435 |
| 4 | * EOF * | | SUM = | 127 |

d. Contents of NEW1 produced by: COPYL(OLD,LGO,NEW1,RA)

| REC | ITEMIZE OF NEW1 NAME | TYPE | FILE LENGTH | 1 CKSUM |
|-----|----------------------|------|-------------|---------|
| 1 | SUBA | REL | 32 | 1435 |
| 2 | SUBB | REL | 40 | 2643 |
| 3 | SUBC | REL | 37 | 5657 |
| 4 | SUBB | REL | 34 | 0727 |
| 5 | SUBD | REL | 40 | 6320 |
| 6 | SUBB | REL | 34 | 0727 |
| 7 | SUBZ | REL | 35 | 2312 |
| 8 | * EOF * | | SUM = | 316 |

f. Contents of NEW3 produced by: COPYLM(NEW3,R)

| REC | ITEMIZE OF NEW3 NAME | TYPE | FILE LENGTH | 1 CKSUM |
|-----|----------------------|------|-------------|---------|
| 1 | SUBA | REL | 32 | 1435 |
| 2 | SUBB | REL | 40 | 2643 |
| 3 | SUBC | REL | 37 | 5657 |
| 4 | SUBB | REL | 40 | 2643 |
| 5 | SUBD | REL | 40 | 6320 |
| 6 | SUBB | REL | 40 | 2643 |
| 7 | * EOF * | | SUM = | 271 |

Figure 2-2. Contents of Files from Figure 2-1

a. Dayfile from job.

```

ATTACH(LIB,UPDLIB,ID=FT)
PF CYCLE NO. = 010
COPYBR(LIB,OLD,2)
REWIND(LI3)
LIBRARY(LI3)
FTN(B=XXX,L=0) ← .043 CP SECONDS COMPILATION TIME.
ITEMIZE(OLD)
ITEMIZE COMPLETE.
ITEMIZE(XXX)
ITEMIZE COMPLETE.
COPYL(OLD,XXX,NEW1)
COPYL DID NOT FIND -- REL / COPYL ←
COPYL COMPLETE.
ITEMIZE(NEW1)
ITEMIZE COMPLETE.
COPYL(OLD,XXX,NEW2,,RT) ←
UPDATED -- ABS / COPYL
COPYL COMPLETE.
ITEMIZE(NEW2)
ITEMIZE COMPLETE.

```

Program COPYL on INPUT file is relocatable, so type REL shows on file XXX itemize.

Record on XXX that does not match name and type on OLD is reported on dayfile. Type indicated is from replacement file.

With T parameter, only the name needs to match. Updated record type is type from old master.

b. Contents of OLD.

| REC | ITEMIZE OF OLD NAME | TYPE | FILE LENGTH | 1 CKSUM |
|-----|---------------------|-------|-------------|---------|
| 1 | ITEMIZE | ABS | 1737 | 2046 |
| 2 | ITEMIZE COPYL | ABS | 1134 | 1350 |
| | COPYL COPYLM | | | |
| | * EOF * | SUM = | 3073 | |

c. Contents of XXX.

| REC | ITEMIZE OF XXX NAME | TYPE | 1 CKSUM |
|-----|---------------------|-------|---------|
| 1 | COPYL | REL | |
| 2 | * EOF * | SUM = | 33 |

Type differs from OLD

d. Contents of NEW1 produced by:
COPYL(OLD,XXX,NEW1)

| REC | ITEMIZE OF NEW1 NAME | TYPE | FILE LENGTH | 1 CKSUM |
|-----|----------------------|-------|-------------|---------|
| 1 | ITEMIZE | ABS | 1737 | 2046 |
| 2 | ITEMIZE COPYL | ABS | 1134 | 1350 |
| | COPYL COPYLM | | | |
| 3 | * EOF * | SUM = | 3073 | |

e. Contents of NEW2 produced by:
COPYL(OLD,XXX,NEW2,,RT)

| REC | ITEMIZE OF NEW2 NAME | TYPE | FILE LENGTH | 1 CKSUM |
|-----|----------------------|-------|-------------|---------|
| 1 | ITEMIZE | ABS | 1737 | 2046 |
| 2 | ITEMIZE COPYL | REL | 33 | 5426 |
| 3 | * EOF * | SUM = | 1772 | |

Figure 2-3. COPYL Example Showing Use of T Parameter

a. Dayfile from job.

```

FTN(L=0,B=OLD) ← Compile 3 subroutines on file OLD.
COMPILING TEST1 Error in TEST2 results in type COS.
TEST1 - .059 SEC 7241 3/4TN
COMPILING TEST2
1 FORTRAN ERRORS IN TEST2
TEST2 - .013 SEC 27692 3/4TN
COMPILING TEST3
TEST3 - .051 SEC 14117 3/4TN
.126 CP SECONDS COMPILATION TIME
FIN(L=0,B=BIN,BL) ← Compile replacement for TEST2 on file BIN.
COMPILING TEST2 No errors result in type REL record.
TEST2 - .046 SEC 14347 3/4TN
.046 CP SECONDS COMPILATION TIME
ITEMIZE(OLD)
ITEMIZE COMPLETE.
ITEMIZE(BIN)
ITEMIZE COMPLETE.
COPYL(OLD,BIN,NEW)
UPDATED -- COS / TEST2 } Type REL replaces type COS on OLD even
COPYL COMPLETE. without T parameter.
ITEMIZE(NEW)
ITEMIZE COMPLETE.
FTN(BL,L=0,B=UPD) ← Compile replacement for TEST3. Errors result
COMPILING TEST3 in type COS for replacement file UPD.
1 FORTRAN ERRORS IN TEST3
TEST3 - .035 SEC 9571 3/4TN
.037 CP SECONDS COMPILATION TIME
COPYL(NEW,UPD,NEWA,,R) } COS type record on UPD is ignored if no T
COPYL COMPLETE. parameter.
ITEMIZE(NEWA)
ITEMIZE COMPLETE.

```

b. Contents of old master file OLD.

c. Contents of replacement file BIN.

| REC | ITEMIZE OF OLD NAME | TYPE | FILE LENGTH | 1 CKSUM |
|-----|---------------------|-------|-------------|---------|
| 1 | TEST1 | REL | 37 | 2335 |
| 2 | TEST2 | COS | 3 | 3433 |
| 3 | TEST3 | REL | 65 | 4111 |
| 4 | * EOF * | SUM = | 127 | |

| REC | ITEMIZE OF BIN NAME | TYPE | FILE LENGTH | 1 CKSUM |
|-----|---------------------|-------|-------------|---------|
| 1 | TEST2 | REL | 43 | 4326 |
| 2 | * EOF * | SUM = | 43 | |

d. Contents of new master file NEW.

e. Contents of new master file NEWA.

| REC | ITEMIZE OF NEW NAME | TYPE | FILE LENGTH | 1 CKSUM |
|-----|---------------------|-------|-------------|---------|
| 1 | TEST1 | REL | 37 | 2335 |
| 2 | TEST2 | REL | 43 | 4326 |
| 3 | TEST3 | REL | 65 | 4111 |
| 4 | * EOF * | SUM = | 167 | |

| REC | ITEMIZE OF NEWA NAME | TYPE | FILE LENGTH | 1 CKSUM |
|-----|----------------------|-------|-------------|---------|
| 1 | TEST1 | REL | 77 | 2375 |
| 2 | TEST2 | REL | 43 | 4326 |
| 3 | TEST3 | REL | 65 | 4111 |
| 4 | * EOF * | SUM = | 167 | |

Figure 2-4. COPYL Example Showing COS Type Record Handling

ITEMIZE lists pertinent information about each record of a binary file in a format suitable for printing. Table 3-1 describes the types of records processed by ITEMIZE.

ITEMIZE processes mass storage files or tape files in a format internal to the operating system (SCOPE format for SCOPE 3.4; I, SI, or X format for KRONOS and NOS; record type S or W for SCOPE 2.1). A file can be processed from beginning-of-information through end-of-information.

Output from ITEMIZE is affected by the type of record and options selected. A header appears for each file terminated by an end-of-file marker (SCOPE 2.1 end-of-partition) within the file specified by logical file name. The first line of the header

identifies the logical file name, file position within that file, and the date and time of the run. The second line of the header has the following fields:

| | |
|--------|---|
| REC | Position of record starting with 1 for each file. |
| NAME | Record name obtained from the second word of the prefix table or from the first word of the record. |
| TYPE | Type of record as shown in table 3-1. |
| LENGTH | Number of words (octal) in the record, excluding the prefix table. |

TABLE 3-1. TYPES OF RECORDS LISTED BY ITEMIZE

| Type of Record | Record Description | Type of Record | Record Description |
|----------------|--|----------------|---|
| ABS | Central processor overlay with one or more named entry points | PPNT | Peripheral processor program name table |
| ACF | MODIFY compressed compile file | REL | Relocatable central processor program |
| COS | Chippewa format central processor program; COMPASS or FORTRAN program with errors that suppressed binaries | SDR | Special deadstart record |
| DATA | Not any other described record type | TEXT | Text record such as a KRONOS procedure file |
| LIBNT | Library name table record | UCF | UPDATE compressed compile file |
| OPL | MODIFY program library deck | ULIB | KRONOS user library header record |
| OPLC | MODIFY program library common deck | UPLx | UPDATE sequential program library with x master control character |
| OPLD | MODIFY program library directory | 6PP | 6000 Series peripheral processor overlay |
| OVL | Central processor overlay with one unnamed entry point (no ENTRY statement in program); system text | 7PP | 7000 Series peripheral processor overlay |

| | |
|----------|--|
| CKSUM | Cyclic logical checksum (octal), excluding the prefix table. |
| DATE | Date record was created, as stored in the prefix table. |
| COMMENTS | Contents of prefix table. |

If no prefix table is present, associated fields are blank.

Additional information listed depends on the type of record:

| | |
|----------------|---|
| ABS | Entry point names are listed. |
| DATA | First line of record is listed if the name of the record is OVERLAY. |
| OPL, OPLC, UPL | Deck names are listed. |
| OVL | Overlay level is listed in octal. |
| TEXT | Entire record is listed if the name of the record is CMRDC, IPRDECK, IPRDC, LIBDECK, LIBDC, or COMMENT. |
| 6PP | Information stored by EDITLIB is listed giving the octal equivalent of the load address, residence, and control card call flag. |
| 7PP | PP number is listed. |

The E option can select further details about several types of records.

The last record in each file is the end-of-file marker, which appears on the listing as the characters * EOF *. The SUM= identification is the total length for all records in the file, including prefix table lengths.

Any zero length record in the file appears with the record name (00). When it is encountered, a sum of the lengths of the records encountered since the beginning of the file, or since the last sum was taken, is listed on the output. The length includes prefix tables. Record numbering is not restarted until a new file is encountered.

SCOPE 3.4 deadstart tapes can be recognized by ITEMIZE. (Deadstart tapes for the other operating systems do not need special processing since prefix

tables exist at the start of each record.) For SCOPE 3.4 deadstart tapes, ITEMIZE lists deadstart records or the library name tables according to their positions on the tape. The remaining records are listed as usual, with the library name becoming part of the header for each file.

A field length of 12000 octal is required for ITEMIZE execution.

A dayfile message is issued when ITEMIZE completes execution.

Control statement format:

ITEMIZE(lfn,p1,p2,...pn)

The first parameter is positional; if omitted, its position must be indicated by a comma. All other parameters are optional and order independent.

| | |
|-----------|--|
| lfn | Logical file name of binary file to be itemized; default name is LGO. |
| L=listlfn | List output on file listlfn; default is L=OUTPUT. |
| BL | Burstable listing; each file output starts at the top of a page. Default is a compact listing in which a page eject occurs only when the current page is nearly full. |
| PD | Print densely at eight lines per inch; default is 6 lines per inch. If this parameter is to produce desired results, the programmer must ensure that output appears at a printer with eight-lines-per-inch capability. |
| NR | No rewind of lfn before or after processing; default is rewind before and after processing. |
| N | Itemize until end-of-information is reached. |
| N=n | Itemize n files, where n is a decimal integer; default is N=1. |
| N=0 | Itemize until an empty file is processed. |
| E | Expand output to list further information; default is no expansion. For type REL, list entry points. |

For types OPL and OPLC, list modification set names and their YANK status.

For type UPL, list correction identifier names.

U Itemize all records within ULIB type records; default is list only the user library directory.

If both E and U are selected for ULIB type records, all records in the library will be itemized; since the records are all type REL, their entry points will be listed.

SCOPE user libraries do not appear as a separate type. When they are processed by ITEMIZE, each record is identified by its type according to table 3-1 categories.

Figure 3-1 shows an example of ITEMIZE output from a KRONOS deadstart tape. The ITEMIZE call was: ITEMIZE(,E,N).

Figure 3-2 shows the results of two ITEMIZE calls; the first listing shows output from the U parameter that requests record information from all records in library; the second listing shows output from use of both the U parameter and E parameter that requests entry points for REL type records.

Figure 3-3 shows ITEMIZE output from a SCOPE 3.4 deadstart tape. Figure 3-4 shows ITEMIZE output from a sequential UPDATE old program library.

| REC | ITEMIZE OF LGO NAME TYPE | FILE LENGTH | 1 CKSUM | DATE /02/06. 12.25.50. |
|--|--|----------------|-------------|--|
| 1 | PRL 6PP (6726) | 160 | 2771 | 74/09/11 73/05 |
| 2 | 2PR 6PP (6606) | 50 | 4064 | 74/ |
| <p>Table information for type 6PP</p> | | | | |
| 13 | CMRDECK TEXT CMRDECK NAME=6400 S/V 127. KRONOS/NOS INTEGRATION. VERSION=KRONOS 2.1-7/386C NCP=27. EQ=DB,ON,1,0,0. | 114 | 7122 | |
| <p>Record contents listed for TEXT when name is CMRDECK</p> | | | | |
| 221 | (00.) | SUM = | 54011 | Zero length record forces total length |
| 222 | PFILES ABS | 1052 | 7113 | 74/09/11 73/05 |
| <p>APPEND ATTACH SAVE REL= SDM=</p> <p>Entry points for type ABS always appear</p> | | | | |
| 223 | PFAITC1 OVL 01,00 | 2130 | 7300 | 74/09/11 73/05 |
| 224 | PFCAT1 OVL 01,00 | 3752 | 4222 | 7 |
| <p>Overlay level</p> | | | | |
| 719 1014 | SYSIO SYSMISC | ULIB ULIB | 12770 41 | 7307 0725 |
| <p>Only directory listed if U parameter omitted</p> | | | | |

Figure 3-1. ITEMIZE Example of Records on KRONOS Deadstart Tape.

a. U parameter itemizes records in library.

| REC | ITEMIZE OF NAME | ULIB TYPE | FILE LENGTH | 1 CKSUM | |
|-----|-----------------|-----------|-------------|---------|----------|
| 1 | ULIB | ULIB | 12 | 6367 | 75/ |
| 2 | LIBREC1 | REL | 17 | 6013 | 75/02 |
| 3 | LIBREC2 | REL | 24 | 3711 | 75/02/ |
| 4 | LIBREC3 | REL | 31 | 6166 | 75/02/ |
| 5 | ULIB | OPLD | 11 | 0535 | 75/02/1. |
| 6 | * EOF * | SUM = | 117 | | |

Records within ULIB listed individually

b. E parameter lists entry points of records listed through U option.

| REC | ITEMIZE OF NAME | ULIB TYPE | FILE LENGTH | 1 CKSUM | |
|-----|-----------------|-----------|-------------|---------|----------|
| 1 | ULIB | ULIB | 12 | 6367 | 75/ |
| 2 | LIBREC1 | REL | 17 | 6013 | 75/0 |
| 3 | LIBREC2 | REL | 24 | 3711 | 75/02. |
| 4 | LIBREC3 | REL | 31 | 6166 | 75/02/ |
| 5 | ULIB | OPLD | 11 | 0535 | 75/02/1. |
| 6 | * EOF * | SUM = | 117 | | |

Entry points within records listed if type is REL

Figure 3-2. ITEMIZE Example Using U and E Parameters

| REC | ITEMIZE OF LGO NAME | TYPE | FILE LENGTH | 1 CKSUM | DATE | COMMENTS |
|--|---------------------|------------|-------------|---------|--------------|------------------|
| 1 | CEA | 6PP (0020) | 6 | 5455 | | |
| 2 | CES | 6PP (0020) | 437 | 2156 | | |
| ≈ | | | | | | |
| 22 | CMRDECK | SDR | 24614 | 3764 | SCOPE 3.4 | SN1 |
| 23 | CMRDECK | SDR | 30261 | 7501 | SCOPE 3.4 | SN74-108 |
| <p style="text-align: center;">Special deadstart record</p> | | | | | | |
| ≈ | | | | | | |
| 46 | DSU | 6PP (0006) | 1446 | 1641 | | |
| 47 | SCOPE 3.4 | LIBNT | 106 | 0530 | LIBRARIES | |
| 48 | SCOPE 3.4 | PPNT | 1564 | 3363 | | |
| <p style="text-align: center;">Library and program name tables</p> | | | | | | |
| ≈ | | | | | | |
| 490 | 900 | 6PP (3611) | 3 | 2735 | 11/21/74 | 02.23.13. |
| 491 | * EOF * | SUM = | 675622 | | | |
| <p style="text-align: center;">Level 17 file terminator summary</p> | | | | | | |
| ≈ | | | | | | |
| REC | ITEMIZE OF LGO NAME | TYPE | FILE LENGTH | 2 CKSUM | LIBRARY DATE | NUCLEUS COMMENTS |
| 1 | ACCOUNT | COS | 372 | 7335 | | |
| 2 | | TEXT | 1657 | 4134 | | |
| <p style="text-align: center;">Library name of following records</p> | | | | | | |

Figure 3-3. ITEMIZE Example of SCOPE 3.4 Deadstart Tape

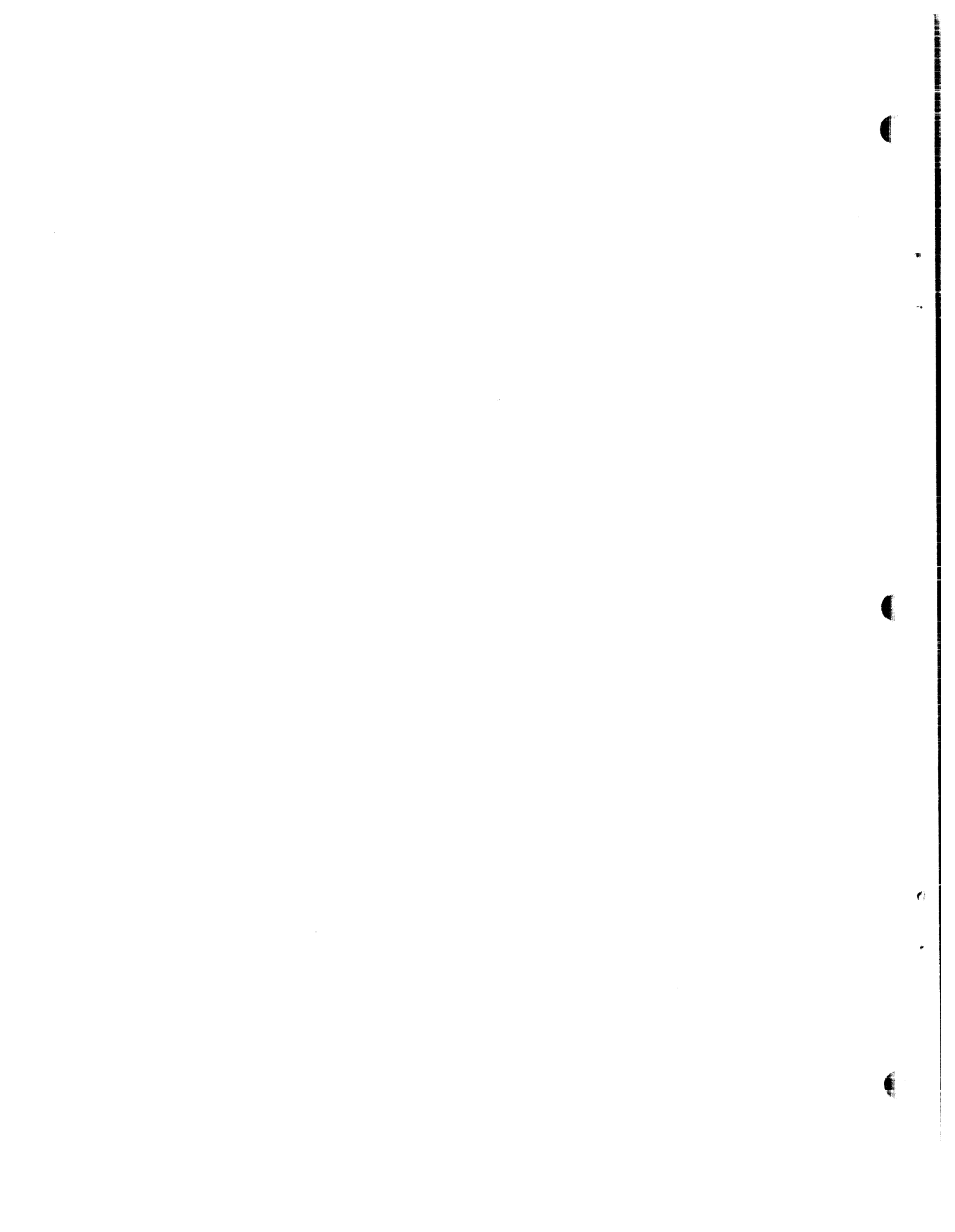
* is control character on OLDPL

| REC | ITEMIZE OF NEWPL NAME | TYPE | FILE LENGTH | 1 CKSUM | DATE | | |
|--|-----------------------|---------|-------------|---------|----------|----------|----------|
| 1 | CHECK 4 | UPL* | 311503 | 5657 | | | |
| CORRECTION IDENTIFIERS LISTED IN ORDER OF INSERTION. | | | | | | | |
| 8-lines per inch listing | UPDATE | SC30007 | SC30014 | SC30045 | SC30046 | SC30047 | SC30052 |
| | SC30059 | SC30060 | SC30067 | SC30068 | SC30075 | ACT30053 | SC30081* |
| | SC30110 | SC30131 | SC30139 | SC30169 | SC30207 | SC30213 | SC30217 |
| | SC30282 | SC30330 | SC30443 | SC30512 | SC30513 | SC30514 | SC30515 |
| | SC30602 | SC30566 | PTR4299 | SC30610 | SC30613 | SC30614 | SC30615 |
| | SC30685 | SC30695 | SC30697 | SC30487 | SC30696 | SC30722 | SC30723 |
| | SC30886 | SC30892 | SC30922 | SC30895 | SC30486 | SC30901 | SC30902 |
| | SC30696A | SC30736 | SC31031 | SC31209 | SC31291 | SC30979 | SC30980 |
| | SC30883 | SC31197 | SC31340 | SC31377 | SC30590 | SC31324 | SC31325 |
| | SC31330 | SC31365 | SC31557 | SC31575 | SC40125A | SC30551 | SC30552 |
| | SC30774A | SC31016 | SC31606 | VER20 | SC40432 | SC31700 | SC31701 |
| | UPD0002 | UPD0006 | UPD0007 | UPD0009 | UPD0030 | UPD0031 | UPD0032 |
| | UPD0029 | UPC0045 | UPD0017 | UPC0020 | UPD0027 | UPD0028 | UPD0029 |
| | UPD0068 | UPD0080 | UPD0031 | UPC0039 | LPD0082 | LPD0083 | LPD0084 |
| | UPD0091 | LPC0094 | UPD0014 | UPC0072 | UPD0087 | UPD0088 | UPD0089 |
| | HISTCRY | UPC0011 | UPD0013 | UPC0061 | UPD0063 | UPD0064 | UPD0065 |
| | L383 | UPD0025 | UPD0050 | UPC0062 | UPD0091A | UPD0091B | UPD0091C |
| | UPD0113 | UPD0117 | ISVL1C | COMAREG | COMA7I0 | COMA7I1 | COMA7I2 |
| | COMCCOD | COMCDAT | COMCDXB | COMCGFL | COMCMVE | COMCMVE | COMCMVE |
| | COMCRDW | COMCSES | COMCSFN | COMCSRT | COMCSYS | COMCSYS | COMCSYS |
| | COMCWTW | COMFGFL | COMFRDC | COMFRIC | COMFRIC | COMFRIC | COMFRIC |
| | COMCSRT | COPYL | ITEMIZE | L390 | L390 | L390 | L390 |
| | UPD0116 | UPC0121 | UPD0128 | UPC0131 | UPC0131 | UPC0131 | UPC0131 |
| | DECK LIST. | | | | | | |
| | | HISTORY | COMAREG | COMA7I0 | COMCCOD | COMCCOD | COMCCOD |
| | COMCDXB | COMCGFL | COMCMVE | COMCMVE | COMCMVE | COMCMVE | |
| | COMCSFN | COMCSRT | COMCSYS | COMCSYS | COMCSYS | COMCSYS | |
| | COMFRDC | COMFRDW | COMFRIC | COMFRIC | COMFRIC | COMFRIC | |
| | ITEMIZE | UPDATE | | | | | |
| 2 | * EOF * | SUM = | 311503 | | | | |

Correction identifiers selected by E parameter.

Deck names listed for type UPL records.

Figure 3-4. ITEMIZE Example of Sequential UPDATE File Records and Dense Listing



STANDARD CHARACTER SETS

A

CONTROL DATA operating systems offer the following variations of a basic character set:

- CDC 64-character set
- CDC 63-character set
- ASCII 64-character set
- ASCII 63-character set

These character sets are listed in table A-1. The set in use at a particular installation was specified when the operating system was installed.

Depending on another installation option, the system assumes an input deck has been punched either in

026 or in 029 mode (regardless of the character set in use). The user, however, may specify the alternate mode by a 26 or 29 punched in columns 79 and 80 of the job card or any 7/8/9 card. The specified mode remains in effect through the end of the job unless it is reset by specification of the alternate mode on a subsequent 7/8/9 card.

Under NOS and KRONOS, the alternate mode can be specified also by a 26 or 29 punched in columns 79 and 80 of any 6/7/9 card, as described above for a 7/8/9 card. In addition, 026 mode can be specified by a card with 5/7/9 multipunched in column 1, and 029 mode can be specified by a card with 5/7/9 multipunched in column 1 and a 9 punched in column 2.

STANDARD CHARACTER SETS

| CDC Graphic | ASCII Graphic Subset | Display Code | Hollerith Punch (026) | External BCD Code | ASCII Punch (029) | ASCII Code | CDC Graphic | ASCII Graphic Subset | Display Code | Hollerith Punch (026) | External BCD Code | ASCII Punch (029) | ASCII Code |
|-------------|----------------------|--------------|-----------------------|-------------------|-------------------|------------|--------------|----------------------|--------------|-----------------------|-------------------|-------------------|------------|
| :t | : | 00† | 8-2 | 00 | 8-2 | 3A | 6 | 6 | 41 | 6 | 06 | 6 | 36 |
| A | A | 01 | 12-1 | 61 | 12-1 | 41 | 7 | 7 | 42 | 7 | 07 | 7 | 37 |
| B | B | 02 | 12-2 | 62 | 12-2 | 42 | 8 | 8 | 43 | 8 | 10 | 8 | 38 |
| C | C | 03 | 12-3 | 63 | 12-3 | 43 | 9 | 9 | 44 | 9 | 11 | 9 | 39 |
| D | D | 04 | 12-4 | 64 | 12-4 | 44 | + | + | 45 | 12 | 60 | 12-8-6 | 28 |
| E | E | 05 | 12-5 | 65 | 12-5 | 45 | - | * | 46 | 11 | 40 | 11 | 2D |
| F | F | 06 | 12-6 | 66 | 12-6 | 46 | * | * | 47 | 11-8-4 | 54 | 11-8-4 | 2A |
| G | G | 07 | 12-7 | 67 | 12-7 | 47 | / | / | 50 | 0-1 | 21 | 0-1 | 2F |
| H | H | 10 | 12-8 | 70 | 12-8 | 48 | (| (| 51 | 0-8-4 | 34 | 12-8-5 | 28 |
| I | I | 11 | 12-9 | 71 | 12-9 | 49 |) |) | 52 | 12-8-4 | 74 | 11-8-5 | 29 |
| J | J | 12 | 11-1 | 41 | 11-1 | 4A | \$ | \$ | 53 | 11-8-3 | 53 | 11-8-3 | 24 |
| K | K | 13 | 11-2 | 42 | 11-2 | 4B | = | = | 54 | 8-3 | 13 | 8-6 | 3D |
| L | L | 14 | 11-3 | 43 | 11-3 | 4C | blank | blank | 55 | no punch | 20 | no punch | 20 |
| M | M | 15 | 11-4 | 44 | 11-4 | 4D | , (comma) | , (comma) | 56 | 0-8-3 | 33 | 0-8-3 | 2C |
| N | N | 16 | 11-5 | 45 | 11-5 | 4E | . (period) | . (period) | 57 | 12-8-3 | 73 | 12-8-3 | 2E |
| O | O | 17 | 11-6 | 46 | 11-6 | 4F | ≡ | # | 60 | 0-8-6 | 36 | 8-3 | 23 |
| P | P | 20 | 11-7 | 47 | 11-7 | 50 | | | 61 | 8-7 | 17 | 12-8-2 | 5B |
| Q | Q | 21 | 11-8 | 50 | 11-8 | 51 | | | 62 | 0-8-2 | 32 | 11-8-2 | 5D |
| R | R | 22 | 11-9 | 51 | 11-9 | 52 | % | % | 63†† | 8-6 | 16 | 0-8-4 | 25 |
| S | S | 23 | 0-2 | 22 | 0-2 | 53 | ≠ | " (quote) | 64 | 8-4 | 14 | 8-7 | 22 |
| T | T | 24 | 0-3 | 23 | 0-3 | 54 | → | _ (underline) | 65 | 0-8-5 | 35 | 0-8-5 | 5F |
| U | U | 25 | 0-4 | 24 | 0-4 | 55 | v | ! (exclamation) | 66 | 11-0 or 11-8-2††† | 52 | 12-8-7 or 11-0††† | 21 |
| V | V | 26 | 0-5 | 25 | 0-5 | 56 | ^ | & | 67 | 0-8-7 | 37 | 12 | 26 |
| W | W | 27 | 0-6 | 26 | 0-6 | 57 | ↑ | ' (apostrophe) | 70 | 11-8-5 | 55 | 8-5 | 27 |
| X | X | 30 | 0-7 | 27 | 0-7 | 58 | ↓ | ? (question) | 71 | 11-8-6 | 56 | 0-8-7 | 3F |
| Y | Y | 31 | 0-8 | 30 | 0-8 | 59 | < | < | 72 | 12-0 or 12-8-2††† | 72 | 12-8-4 or 12-0††† | 3C |
| Z | Z | 32 | 0-9 | 31 | 0-9 | 5A | > | > | 73 | 11-8-7 | 57 | 0-8-6 | 3E |
| 0 | 0 | 33 | 0 | 12 | 0 | 30 | > | @ | 74 | 8-5 | 15 | 8-4 | 40 |
| 1 | 1 | 34 | 1 | 01 | 1 | 31 | ≈ | ~ (circumflex) | 75 | 12-8-5 | 75 | 0-8-2 | 5C |
| 2 | 2 | 35 | 2 | 02 | 2 | 32 | ≡ | ;(semicolon) | 76 | 12-8-6 | 76 | 11-8-7 | 5E |
| 3 | 3 | 36 | 3 | 03 | 3 | 33 | ;(semicolon) | ;(semicolon) | 77 | 12-8-7 | 77 | 11-8-6 | 3B |
| 4 | 4 | 37 | 4 | 04 | 4 | 34 | | | | | | | |
| 5 | 5 | 40 | 5 | 05 | 5 | 35 | | | | | | | |

† Twelve or more zero bits at the end of a 60-bit word are interpreted as end-of-line mark rather than two colons. End-of-line mark is converted to external BCD 1632.

†† In installations using the CDC 63-graphic set, display code 00 has no associated graphic or Hollerith code; display code 63 is the colon (8-2 punch).

††† The % graphic does not exist and translations from ASCII/EBCDIC % yield a blank (55g).

††† The alternate Hollerith (026) and ASCII (029) punches are accepted for input only.

DIAGNOSTIC MESSAGES

B

Diagnostic messages for the utilities are listed alphabetically by their first word. Each message is identified with its issuing routine, significance, and action to be taken.

Output code in table B-1 is as follows: B for B-display, S for system dayfile and J for job dayfile.

TABLE B-1. DIAGNOSTIC MESSAGES

| Message | Significance | Action | Output | Issuing Routine |
|--|---|---|---------|-----------------|
| APPENDED -- type/name | The record with type and name on the replacement file was not matched; it has been appended to the new master file. A option selected. | Informative message. | S, J, B | COPYL/COPYLM |
| COPYING -- type/name | The record with the type and name on the old master file was copied to the new master file. | Informative message. | B | COPYL/COPYLM |
| COPYL COMPLETE | All records on the old master file have been processed. | Informative message. | S, J, B | COPYL/COPYLM |
| COPYL DID NOT FIND type/name | The record with type and name on the replacement file was not found on the old master file. Since the A option was not selected, the record is ignored. | Informative message. | S, J, B | COPYL/COPYLM |
| ERROR IN ARGUMENTS. | Fatal error; illegal argument was specified. Any EXIT(S) processing occurs. | Correct control statement. | S, J, B | ITEMIZE |
| ERROR IN 5TH COPYL PARAMETER | Fatal error; only A, R, and T are recognized parameters. Any EXIT(S) processing occurs. | Correct processing option parameter. | S, J, B | COPYL/COPYLM |
| FILE NAME CONFLICT. | Fatal error; output file name and binary file name were the same. Any EXIT(S) processing occurs. | Rename either output file or file to be itemized. | S, J, B | ITEMIZE |
| ILLEGAL FILE COUNT. | Fatal error; the file count was not numeric. Any EXIT(S) processing occurs. | Correct the N=n parameter. | S, J, B | ITEMIZE |
| ITEMIZE COMPLETE. | Specified processing is finished. | Informative message. | S, J, B | ITEMIZE |
| ITEMIZING xxxx | Record identified is being processed. | Informative message. | B | ITEMIZE |
| OLD MASTER FILE EMPTY OR MISPOSITIONED | An informative diagnostic. The contents of the replacement file are given in the format: COPYL DID NOT FIND -- type/name | Informative message. | S, J, B | COPYL/COPYLM |
| UPDATED -- type/name | The record with type and name on the old master file was replaced with the matching record from the replacement file. | Informative message. | S, J, B | COPYL/COPYLM |

COMMENT SHEET



TITLE: CYBER Common Utilities Reference Manual

PUBLICATION NO. 60493300 REVISION A

This form is not intended to be used as an order blank. Control Data Corporation solicits your comments about this manual with a view to improving its usefulness in later editions.

Applications for which you use this manual.

Do you find it adequate for your purpose?

What improvements to this manual do you recommend to better serve your purpose?

Note specific errors discovered (please include page number reference).

CUT ON THIS LINE

General comments:

FROM NAME: _____ POSITION: _____

COMPANY NAME: _____

ADDRESS: _____

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOI D ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
 PERMIT NO. 8241
 MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
 NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION

Publications and Graphics Division

215 Moffett Park Drive

Sunnyvale, California 94086



CUT ON THIS LINE

FOLD

FOLD

STAPLE

STAPLE





CORPORATE HEADQUARTERS, 8100 34th AVE. SO., MINNEAPOLIS, MINN. 55448
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.



**CYBER COMMON UTILITIES
REFERENCE MANUAL**

**CDC® OPERATING SYSTEMS:
NOS 1
NOS/BE 1
SCOPE 2**

As part of Control Data's continuing quality improvement program, we invite you to complete this questionnaire so that you may have a more direct influence on the manuals you use.

Please rate this manual for each general and individual category on a scale of 1 through 5 as follows:

1 - Excellent 2 - Good 3 - Fair 4 - Poor 5 - Unacceptable

I. Writing Quality

- A. Technical accuracy
- B. Completeness
- C. Audience defined properly
- D. Readability
- E. Understandability
- F. Organization

II. Examples

- A. Quantity
- B. Placement
- C. Applicability
- D. Quality
- E. Instructiveness

III. Format

- A. Type size
- B. Page density
- C. Art work
- D. Legibility
- E. Printing/Reproduction

IV. Miscellaneous

- A. Index
- B. Glossary

V. Please provide a yes or no answer regarding manuals in general:

- A. I prefer that a manual on a software product be as comprehensive as possible; physical size is of little importance.
- B. I prefer that information on a software product be covered in several small manuals, each covering a certain aspect of the product. Smaller manuals with limited subject matter are easier to work with.
- C. I am interested primarily in reference manuals designed for ease of locating specific information.

D. I am interested primarily in user guides designed to teach the user about a product or certain capabilities of a product.

VI. We recognize that we have a wide variety of users. Please identify your primary area of interest or activity:

- A. Student
- B. Applications programmer
- C. Systems programmer
- D. How many years programming experience do you have?
- E. What languages
 - 1. Algol
 - 2. Basic
 - 3. Cobol
 - 4. Compass
 - 5. Fortran
 - 6. PL/I
 - 7. Other

F. Have you ever worked on non-CDC equipment?

- 1. If yes, approximately what percent of your experience is on non-CDC equipment?
- 2. How do you rate CDC manuals against other similar manuals using the 1-5 ratings. (Example: XYZ Corp. 2 means XYZ manuals are good as compared to CDC manuals.)
 - Burroughs
 - DEC
 - Hewlett-Packard
 - Honeywell
 - IBM
 - NCR
 - Univac
 - Other _____

General Comments _____

STAPLE

STAPLE

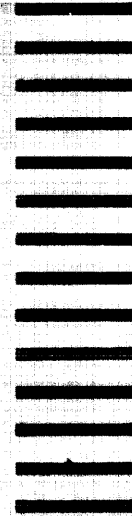
FOLD

FOLD

FIRST CLASS
 PERMIT NO. 8241
 MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
 NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY
CONTROL DATA CORPORATION
Publications and Graphics Division
215 Moffett Park Drive
Sunnyvale, California 94086



CUT ON THIS LINE

FOLD

FOLD

STAPLE

STAPLE



PREFACE

This publication describes the utilities that operate under the following operating systems:

NOS 1 for the CONTROL DATA® CYBER 170 Models 171, 172, 173, and 174, CYBER 70 Models 71, 72, 73, 74, and 6000 Series Computer Systems.

NOS/BE 1 for the CDC® CYBER 170, Series CYBER 70 Models 71, 72, 73, 74, and 6000 Series Computer Systems.

SCOPE 2 for the CONTROL DATA CYBER 170 Model 176, CYBER 70, Model 76 and 7600 Computer Systems.

The utilities described are:

- COPYL/ Maintains a single binary file.
- COPYLM (Replaces COPYLM utility under SCOPE 2.1.)
- ITEMIZE Lists information about the contents of a binary file or multifile.

The reader is assumed to be familiar with the operating system on which the utility is to be run as well as CYBER Record Manager or Record Manager features.

Other utility programs are documented in their respective operating system manuals.

Other documents of interest:

| <u>Publication</u> | <u>Publications Number</u> |
|-----------------------------------|----------------------------|
| NOS 1 Reference Manual (Volume 1) | 60435400 |
| NOS/BE 1 Reference Manual | 60493800 |
| SCOPE 2 Reference Manual | 60342600 |

CDC manuals can be ordered from Control Data Corporation Literature and Distribution Services, 8001 East Bloomington Freeway, Minneapolis, Mn 55420.

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or parameters.



The terminology used for files and records differs somewhat within the different environments of the various operating systems. Generally, terms used with NOS 1 and NOS/BE 1 are similar; terms used with SCOPE 2.1 are different, reflecting SCOPE 2.1 use of Record Manager within the operating system.

The physical representation of files and records differs among operating systems and among different devices within the same operating system. The logical interpretation of physical entities varies less. The CYBER utilities recognize logical structures applicable to all the operating systems.

Terms applicable to this manual and the common CYBER utilities are discussed below. For a complete discussion of possible file formats, consult the reference manuals for the respective operating systems.

FILE STRUCTURE DEFINITIONS

A file is that collection of information known to an operating system by a logical file name. A file begins at beginning-of-information and ends at end-of-information. No user information exists outside these boundaries.

A division smaller than beginning-of-information to end-of-information is known by several different names, depending on the operating system.

Under SCOPE 2.1, information between beginning-of-information and end-of-information can be divided into partitions which, in turn, can be divided into sections. The term record is restricted to a data grouping in a format specified to Record Manager by an RT mnemonic parameter in a file information table in memory or by a FILE control statement. Examples of Record Manager record types are W, in which a control word provides record length, and F, which is a fixed length record.

Under NOS 1 and NOS/BE 1, information between beginning-of-information and end-of-information is

divided into logical records terminated by a marker containing a level number. A marker containing a level 17₈ establishes the next smaller division between beginning- and end-of-information; the level 17₈ marker is commonly termed an end-of-file marker. A record terminated with a level 0 marker is smaller than a level 17₈ record; the level 0 marker is commonly termed an end-of-record marker. The physical file formats that produce the logical end-of-file and end-of-record are the formats internal to the operating system (NOS 1 SI (SCOPE internal) or I (internal), tape format or mass storage; NOS/BE 1 mass storage and tape system-logical-record format).

Terms logically equivalent among operating systems are:

| <u>SCOPE 2.1</u> | <u>NOS 1, NOS/BE 1</u> |
|------------------|---|
| file | file |
| partition | level 17 ₈ record; also referred to as a file |
| section | level 0 record; also referred to as a record |

In this manual, terms from NOS 1 and NOS/BE 1 are used predominantly. As a result, the SCOPE 2.1 end-of-partition becomes an end-of-file for the purposes of many discussions, and the SCOPE 2.1 section is referred to as a record.

TYPES AND NAMES OF RECORDS

The type and the name of a record are determined by the COPYL and ITEMIZE utilities from information contained within the record. If the record begins with a prefix table, the record name is obtained from that table and the type of the record is determined from the first word following the prefix table. If the first word in the record is not a prefix table, but is a recognizable format, the format determines type. Any record that has neither a prefix table nor a recognizable format is classed as a DATA type record.

PREFIX TABLE USE

Prefix tables exist, unless they have been specifically suppressed, for:

Programs assembled or compiled under any operating system

System text overlays

NOS 1 user library header records and directory records

MODIFY program library decks, common decks, or directory records

The prefix table is the first of the ordered set of binary tables that form object programs. The tables consist of a header word with an octal table type identifier followed by varying amounts of control information that instruct system routines such as the loader or that contain the program code.

The prefix table is identified by octal digits 7700 in bits 48-59 of its first word; consequently, it is often referred to as a 77 or 7700 table. Information in the prefix table, which originates with the assembler or other system routine that creates the table, specifies items such as the date created and the system on which the job was executed.

Although some of the records may contain display coded data (loader directives, for instance, are coded), they are considered binary records.

OTHER RECORD IDENTIFIERS

If a prefix table is not present, the first word in a record is examined in a search for a recognizable format. An UPDATE sequential program library, for example, is identified by the characters CHECK in the word.

If a record meets the criteria for a given type of record, the utilities identify it as such. For instance, a load file beginning with a job statement may be identified as type COS, TEXT, or DATA, depending on the particular characters in the job statement.

Table 1-1 summarizes types of records and the criteria used to determine them.

RECORD NAMES

If a record begins with a prefix table, bits 18-59 of the second word of the table determine the record name. If a record does not begin with a prefix table, bits 18-59 of the first word of the record are used as the record name.

Records typed as DATA, ACF, UCF, and UPL do not have names.

COPYL copies an old master file to a new master file, substituting records from a replacement file for matching records on the old master, and optionally adding records to the end of the new master file. Records are considered to match if they have the same type and name, or optionally, the same name. COPYL operates with binary or text records. It is commonly used to maintain files of procedures or subroutines.

COPYL and COPYLM differ only in the handling of multiple occurrences of a record on the old master. COPYL uses each record on the replacement file only once, replacing the first matching record from the old master file. COPYLM uses the first matching record encountered on the replacement file to replace each matching record from the old master file. COPYL can be used for multiple replacement only if multiple copies of the record are on the replacement file.

The master and replacement files must reside on mass storage or a binary tape in a format internal to the operating system (system-logical-record format tape for NOS/BE 1; I or SI format for NOS 1; record type S or W for SCOPE 2.1). Only a single file terminated by a level 17₈ record marker (SCOPE 2.1 partition) is processed by a single call to COPYL or COPYLM.

Order of records on the replacement file is not significant. Records on the new master file are in the same order as they were on the old master file.

COPYL issues dayfile messages (appendix B) during processing; no other printed output is produced.

A field length of 12000 octal is required for execution.

COPYL replaces only the types of records listed in table 2-1. Any record on the old master file that is not recognized as one of the listed types is copied to the new master file without further processing.

TABLE 2-1. TYPES[†] OF RECORDS REPLACED BY COPYL AND COPYLM

| Type | Description |
|------|--|
| ABS | Central processor overlay with one or more named entry points |
| CAP | Capsule |
| COS | Chippewa format central processor program, FORTRAN binary with compilation errors or COMPASS binary with assembly errors |
| OPL | MODIFY old program library deck |
| OPLC | MODIFY old program library common deck |
| OVL | Central processor overlay with one unnamed entry point including system texts |
| PROC | CYBER CONTROL LANGUAGE procedure file |
| REL | Relocatable central processor program |
| TEXT | Text record (e.g., NOS 1 procedure) |
| 6PP | 6000 Series peripheral processor program |
| 7PP | 7000 Series peripheral processor program |

[†] For additional information about how these types are determined, consult table 1-1.

Control statement format:

COPYL(oldlfn, replfn, newlfn, last, flag) Single replacement

COPYLM(oldlfn, replfn, newlfn, last, flag) Multiple replacement

All parameters are optional and position dependent:

- oldlfn Logical file name of the old master file; default name is OLD.
- replfn Logical file name of the replacement file; default name is LGO.
- newlfn Logical file name of the updated master file; default name is NEW.
- last Name of the last record on oldlfn to be processed. If last is not specified, all records on oldlfn are processed.
- flag Processing options; default is do not select option.
 - R Rewind oldlfn and newlfn before processing. (Replfn is always rewound before and after processing.)
 - A Append to the end of newlfn all replfn records that do not match any on the oldlfn.
 - T Omit check for matching type of record.

These options can be specified by combining one or more letters in any order, as TRA, AR, RT, or TR.

The R option affects master file positioning only before processing. If R is specified, both the old and new master files are rewound to beginning-of-information before processing. In the absence of R, the user is responsible for positioning these files. After processing, the old master file is positioned after the end-of-file (end-of-partition) that stopped processing. COPYL writes an end-of-file and does not further position the new master file. The R option does not affect the file of replacement records, since the replacement file always is rewound before and after processing.

The A option causes unmatched records from the replacement file to be copied to the new master file following the last record copied from the old master file. If A is not selected, records on the replacement file are ignored and an informative message is issued when they do not match records on the existing file.

The T option determines whether the record name, or name and type, will be used to identify a matching record. If T is not selected, COPYL considers records to match only if both the type and name are the same. If T is selected, however, COPYL considers records to match if they have the same name, regardless of type. Otherwise, any replacement file record that is not of a type listed in table 2-1 is ignored without comment.

Records of type COS (such as result from a compilation or assembly of programs containing errors) are handled in a special manner. If a COS type record appears on a replacement file, it is always ignored without comment. If a COS type record appears on the old master file and a record with the same name but a type other than COS appears on the replacement file, the replacement record with the new type is always copied to the new master file. The match on type is suppressed for COS type records from the old master file even when the T option is not specified. A COS type record on the old master file that is not replaced is copied to the new master file without change.

Figure 2-1 shows the dayfile from a job that produces four new files (NEW, NEW1, NEW2, and NEW3), using the same old master file (OLD) and the same replacement file (LGO). Figure 2-2 identifies the contents of OLD and LGO (figure 2-2a-b) and the newly created files using the ITEMIZE utility.

Figure 2-3 shows an example of the T parameter of COPYL. The old master file (figure 2-3b) contains two absolute records named COPYL and ITEMIZE, as shown by the execution of the ITEMIZE utility. A new relocatable record with the name COPYL is created on replacement file XXX (figure 2-3c). Executing the COPYL utility without the T parameter produces a new master file, NEW1, that is identical to OLD (figure 2-3d). When the T parameter is used, records are replaced if the name matches even though the type may differ, so that NEW2 (figure 2-3e) is not the same as OLD.

Figure 2-4 shows that the T parameter is not required to suppress record type checking when a COS type record on the old master file is to be replaced. A COS type record is presumed to be a record with compilation or assembly errors when a record with a matching name appears on the replacement file. A COS type record on the replacement file does not suppress the type check, however, and in the absence of the T parameter a COS type record cannot be written to the new master file.

ITEMIZE lists pertinent information about each record of a binary file in a format suitable for printing. Table 3-1 describes the types of records processed by ITEMIZE.

ITEMIZE processes mass storage files or tape files in a format internal to the operating system (system logical-record format for NOS/BE 1; I or SI format for NOS 1; record type S or W for SCOPE 2.1). A file can be processed from beginning-of-information through end-of-information.

Output from ITEMIZE is affected by the type of record and options selected. A header appears for each file terminated by an end-of-file marker (SCOPE 2.1 end-of-partition) within the file specified by logical file name. The first line of the header

identifies the logical file name, file position within that file, and the date and time of the run. The second line of the header has the following fields:

| | |
|--------|---|
| REC | Position of record starting with 1 for each file. |
| NAME | Record name obtained from the second word of the prefix table or from the first word of the record. |
| TYPE | Type of record as shown in table 3-1. |
| LENGTH | Number of words (octal) in the record, excluding the prefix table. |

TABLE 3-1. TYPES OF RECORDS LISTED BY ITEMIZE

| Type of Record | Record Description | Type of Record | Record Description |
|----------------|--|----------------|---|
| ABS | Central processor overlay with one or more named entry points | PPNT | Peripheral processor program name table |
| ACF | MODIFY compressed compile file | PROC | CYBER CONTROL LANGUAGE procedure file |
| CAP | Capsule | REL | Relocatable central processor program |
| COS | Chippewa format central processor program; COMPASS or FORTRAN program with errors that suppressed binaries | SDR | Special deadstart record |
| DATA | Not any other described record type | TEXT | Text record such as a NOS 1 procedure file |
| LIBNT | Library name table record | UCF | UPDATE compressed compile file |
| OPL | MODIFY program library deck | ULIB | NOS 1 user library header record |
| OPLC | MODIFY program library common deck | UPLx | UPDATE sequential program library with x master control character |
| OPLD | MODIFY program library directory | 6PP | 6000 Series peripheral processor overlay |
| OVL | Central processor overlay with one unnamed entry point (no ENTRY statement in program); system text | 7PP | 7000 Series peripheral processor overlay |

| | |
|----------|--|
| CKSUM | Cyclic logical checksum (octal), excluding the prefix table. |
| DATE | Date record was created, as stored in the prefix table. |
| COMMENTS | Contents of prefix table. |

If no prefix table is present, associated fields are blank.

Additional information listed depends on the type of record:

| | |
|----------------|---|
| ABS | Entry point names are listed. |
| DATA | First line of record is listed if the name of the record is OVERLAY. |
| OPL, OPLC, UPL | Deck names are listed. |
| OVL | Overlay level is listed in octal. |
| TEXT | Entire record is listed if the name of the record is CMRDC, IPRDECK, IPRDC, LIBDECK, LIBDC, or COMMENT. |
| 6PP | Information stored by EDITLIB is listed giving the octal equivalent of the load address, residence, and control card call flag. |
| 7PP | PP number is listed. |

The E option can select further details about several types of records.

The last record in each file is the end-of-file marker, which appears on the listing as the characters * EOF *. The SUM= identification is the total length, in words, for all records in the file, including prefix table lengths.

Any zero length record in the file appears with the record name (00). When it is encountered, a sum of the lengths of the records encountered since the beginning of the file, or since the last sum was taken, is listed on the output. The length includes prefix tables. Record numbering is not restarted until a new file is encountered.

If a record of type UPL has more correction identifier names and/or deck names than can be accommodated within the ITEMIZE buffer, the following message appears on the listing in place of the excess names: TRUNCATED -- IDENT OR DECK LIST TOO LONG.

In this instance, the UPDATE utility must be used to obtain a complete list of identifiers and deck names.

NOS/BE 1 deadstart tapes can be recognized by ITEMIZE. (Deadstart tapes for the other operating systems do not need special processing since prefix tables exist at the start of each record.) For NOS/BE 1 deadstart tapes, ITEMIZE lists deadstart records or the library name tables according to their positions on the tape. The remaining records are listed as usual, with the library name becoming part of the header for each file.

A field length of 20000 octal is required for ITEMIZE execution.

A dayfile message is issued when ITEMIZE completes execution.

Control statement format:

```
ITEMIZE(lfn,p1,p2,...pn)
```

The first parameter is positional; if omitted, its position must be indicated by a comma. All other parameters are optional and order independent.

lfn Logical file name of binary file to be itemized; default name is LGO.

L=listfn List output on file listfn; default is L=OUTPUT.

BL Burstable listing; each file output starts at the top of a page. Default is a compact listing in which a page eject occurs only when the current page is nearly full.

PW Print width; print 72-character lines regardless of the listing file device.

PW=n Print either 136-character lines or 72-character lines depending on the value of n, where n is a decimal integer. If $n \geq 136$, print 136-character lines. If $n < 136$, print 72-character lines.

If PW is omitted, the default value is 72-character lines if the listing file is a terminal; otherwise, the default value is 136-character lines.

PD Print densely at eight lines per inch; default is 6 lines per inch. If this parameter is to produce desired results, the programmer must ensure that output appears at a printer with eight-lines-per inch capability.

a. U parameter itemizes records in library.

| REC | ITEMIZE OF NAME | ULIB TYPE | FILE LENGTH | 1 CKSUM | |
|-----|-----------------|-----------|-------------|---------|---------|
| 1 | ULIB | ULIB | 12 | 6367 | 75/ |
| 2 | LIBREC1 | REL | 17 | 6013 | 75/02 |
| 3 | LIBREC2 | REL | 24 | 3711 | 75/02/ |
| 4 | LIBREC3 | REL | 31 | 6166 | 75/02/ |
| 5 | ULIB | OPLO | 11 | 0535 | 75/02/1 |
| 6 | * EOF * | SUM = | 117 | | |

Records within ULIB listed individually

b. E parameter lists entry points of records listed through U option.

| REC | ITEMIZE OF NAME | ULIB TYPE | FILE LENGTH | 1 CKSUM | |
|-----|-----------------|-----------|-------------|---------|---------|
| 1 | ULIB | ULIB | 12 | 6367 | 75/ |
| 2 | LIBREC1 | REL | 17 | 6013 | 75/0 |
| 3 | LIBREC2 | REL | 24 | 3711 | 75/02 |
| 4 | LIBREC3 | REL | 31 | 6166 | 75/02/ |
| 5 | ULIB | OPLO | 11 | 0535 | 75/02/1 |
| 6 | * EOF * | SUM = | 117 | | |

Entry points within records listed if type is REL

Figure 3-2. ITEMIZE Example Using U and E Parameters

| REC | ITEMIZE OF TAPE NAME | TYPE | FILE LENGTH | FILE 1 CKSUM | DATE | ** UNLABELED ** COMMENTS |
|-----|----------------------|------------|-------------|--------------|----------|--------------------------|
| 1 | CEA | 6PP (0020) | 6 | 5375 | | |
| 2 | CES | 6PP (0020) | 1274 | 1407 | | |
| 3 | DDT | 6PP (0000) | 1251 | 0320 | 04/21/76 | 09.0' |
| • | | | | | | |
| • | | | | | | |
| • | | | | | | |
| 24 | CMRDC | SDR | 26574 | 1161 | | NOS/BE 1 |
| 25 | CMRDC | SDR | 23653 | 5225 | | NOS/BE 1.1 |
| 26 | CMRDC | SDR | 26424 | 3437 | | NOS/BE 1.1 |
| 27 | CMRDC | SDR | 24607 | 4057 | | NOS/BE 1.1 |
| • | | | | | | |
| • | | | | | | |
| • | | | | | | |
| 496 | 909 | 6PP (6123) | 56 | 4722 | 04/21/76 | 09.08.25. NO |
| 497 | * EOF * | SUM = | 751677 | | | |

Special deadstart record

Level 17₈ file terminator summary

| REC | ITEMIZE OF TAPE NAME | TYPE | FILE LENGTH | FILE 2 CKSUM | LIBRARY DATE | NUCLEUS COMMENTS |
|-----|----------------------|-----------|-------------|--------------|--------------|-------------------------------|
| 1 | ACCOUNT | COS | 426 | 6664 | | |
| 2 | | DATA | 1736 | 7566 | | |
| • | | | | | | |
| • | | | | | | |
| • | | | | | | |
| 12 | CPCTEXT | OVL 01,01 | 4534 | 3544 | 09/21/76 | 15.42.40. SYSTEM TEXT FOR CP' |
| 13 | CPUTEXT | OVL 01,01 | 4026 | 2522 | 09/21/76 | 15.42. SYSTEM TEXT FO' |

Library name of following records

Figure 3-3. ITEMIZE Example of NOS/BE 1 Deadstart Tape

STANDARD CHARACTER SETS

A

CONTROL DATA operating systems offer the following variations of a basic character set:

CDC 64-character set

CDC 63-character set

ASCII 64-character set

ASCII 63-character set

The set in use at a particular installation was specified when the operating system was installed.

Depending on another installation option, the system assumes an input deck has been punched either in 026 or in 029 mode (regardless of the character set in use). Under NOS/BE 1 the alternate mode can be specified by a 26 or 29 punched in columns 79 and 80 of the job statement or any 7/8/9 card. The specified mode

remains in effect through the end of the job unless it is reset by specification of the alternate mode on a subsequent 7/8/9 card.

Under NOS 1, the alternate mode can be specified by a 26 or 29 punched in columns 79 and 80 of any 6/7/9 card, as described above for a 7/8/9 card. In addition, 026 mode can be specified by a card with 5/7/9 multipunched in column 1, and 029 mode can be specified by a card with 5/7/9 multipunched in column 1 and a 9 punched in column 2.

Graphic character representation appearing at a terminal or printer depends on the installation character set and the terminal type. Characters shown in the CDC Graphic column of the standard character set table are applicable to BCD terminals: ASCII graphic characters are applicable to ASCII-CRT and ASCII-TTY terminals.

TABLE A-1. STANDARD CHARACTER SETS

| Display Code (octal) | CDC | | | ASCII | | |
|----------------------|-------------------------|-------------------------------|-------------------|-------------------------|-------------------------------|--------------|
| | Graphic | Hollerith Punch (026) | External BCD Code | Graphic Subset | Punch (029) | Code (octal) |
| 00 [†] | : (colon) ^{††} | 8-2 | 00 | : (colon) ^{††} | 8-2 | 072 |
| 01 | A | 12-1 | 61 | A | 12-1 | 101 |
| 02 | B | 12-2 | 62 | B | 12-2 | 102 |
| 03 | C | 12-3 | 63 | C | 12-3 | 103 |
| 04 | D | 12-4 | 64 | D | 12-4 | 104 |
| 05 | E | 12-5 | 65 | E | 12-5 | 105 |
| 06 | F | 12-6 | 66 | F | 12-6 | 106 |
| 07 | G | 12-7 | 67 | G | 12-7 | 107 |
| 10 | H | 12-8 | 70 | H | 12-8 | 110 |
| 11 | I | 12-9 | 71 | I | 12-9 | 111 |
| 12 | J | 11-1 | 41 | J | 11-1 | 112 |
| 13 | K | 11-2 | 42 | K | 11-2 | 113 |
| 14 | L | 11-3 | 43 | L | 11-3 | 114 |
| 15 | M | 11-4 | 44 | M | 11-4 | 115 |
| 16 | N | 11-5 | 45 | N | 11-5 | 116 |
| 17 | O | 11-6 | 46 | O | 11-6 | 117 |
| 20 | P | 11-7 | 47 | P | 11-7 | 120 |
| 21 | Q | 11-8 | 50 | Q | 11-8 | 121 |
| 22 | R | 11-9 | 51 | R | 11-9 | 122 |
| 23 | S | 0-2 | 22 | S | 0-2 | 123 |
| 24 | T | 0-3 | 23 | T | 0-3 | 124 |
| 25 | U | 0-4 | 24 | U | 0-4 | 125 |
| 26 | V | 0-5 | 25 | V | 0-5 | 126 |
| 27 | W | 0-6 | 26 | W | 0-6 | 127 |
| 30 | X | 0-7 | 27 | X | 0-7 | 130 |
| 31 | Y | 0-8 | 30 | Y | 0-8 | 131 |
| 32 | Z | 0-9 | 31 | Z | 0-9 | 132 |
| 33 | 0 | 0 | 12 | 0 | 0 | 060 |
| 34 | 1 | 1 | 01 | 1 | 1 | 061 |
| 35 | 2 | 2 | 02 | 2 | 2 | 062 |
| 36 | 3 | 3 | 03 | 3 | 3 | 063 |
| 37 | 4 | 4 | 04 | 4 | 4 | 064 |
| 40 | 5 | 5 | 05 | 5 | 5 | 065 |
| 41 | 6 | 6 | 06 | 6 | 6 | 066 |
| 42 | 7 | 7 | 07 | 7 | 7 | 067 |
| 43 | 8 | 8 | 10 | 8 | 8 | 070 |
| 44 | 9 | 9 | 11 | 9 | 9 | 071 |
| 45 | + | 12 | 60 | + | 12-8-6 | 053 |
| 46 | * | 11 | 40 | * | 11 | 055 |
| 47 | | 11-8-4 | 54 | | 11-8-4 | 052 |
| 50 | / | 0-1 | 21 | / | 0-1 | 057 |
| 51 | (| 0-8-4 | 34 | (| 12-8-5 | 050 |
| 52 |) | 12-8-4 | 74 |) | 11-8-5 | 051 |
| 53 | \$ | 11-8-3 | 53 | \$ | 11-8-3 | 044 |
| 54 | = | 8-3 | 13 | = | 8-6 | 075 |
| 55 | blank | no punch | 20 | blank | no punch | 040 |
| 56 | , (comma) | 0-8-3 | 33 | , (comma) | 0-8-3 | 054 |
| 57 | . (period) | 12-8-3 | 73 | . (period) | 12-8-3 | 056 |
| 60 | ≡ | 0-8-6 | 36 | # | 8-3 | 043 |
| 61 | [| 8-7 | 17 | [| 12-8-2 | 133 |
| 62 |] | 0-8-2 | 32 |] | 11-8-2 | 135 |
| 63 | % ^{††} | 8-6 | 16 | % ^{††} | 0-8-4 | 045 |
| 64 | ⌘ | 8-4 | 14 | " (quote) | 8-7 | 042 |
| 65 | ⌞ | 0-8-5 | 35 | _ (underline) | 0-8-5 | 137 |
| 66 | ⌟ | 11-0 or 11-8-2 ^{†††} | 52 | ! | 12-8-7 or 11-0 ^{†††} | 041 |
| 67 | ⌠ | 0-8-7 | 37 | & | 12 | 046 |
| 70 | ⌡ | 11-8-5 | 55 | ' (apostrophe) | 8-5 | 047 |
| 71 | ⌢ | 11-8-6 | 56 | ? | 0-8-7 | 077 |
| 72 | < | 12-0 or 12-8-2 ^{†††} | 72 | < | 12-8-4 or 12-0 ^{†††} | 074 |
| 73 | > | 11-8-7 | 57 | > | 0-8-6 | 076 |
| 74 | ⌥ | 8-5 | 15 | @ | 8-4 | 100 |
| 75 | ⌦ | 12-8-5 | 75 | \ | 0-8-2 | 134 |
| 76 | ⌧ | 12-8-6 | 76 | ˘ (circumflex) | 11-8-7 | 136 |
| 77 | ; | 12-8-7 | 77 | ; | 11-8-6 | 073 |

[†] Twelve zero bits at the end of a 60-bit word in a zero byte record are an end of record mark rather than two colons.
^{††} In installations using a 63-graphic set, display code 00 has no associated graphic or card code; display code 63 is the colon (8-2 punch). The % graphic and related card codes do not exist and translations yield a blank (55g).
^{†††} The alternate Hollerith (026) and ASCII (029) punches are accepted for input only.

COMMENT SHEET



TITLE: CYBER Common Utilities Reference Manual

PUBLICATION NO. 60495600 REVISION E

This form is not intended to be used as an order blank. Control Data Corporation solicits your comments about this manual with a view to improving its usefulness in later editions.

Applications for which you use this manual.

Do you find it adequate for your purpose?

What improvements to this manual do you recommend to better serve your purpose?

Note specific errors discovered (please include page number reference).

CUT ON THIS LINE

General comments:

FROM NAME: _____ POSITION: _____

COMPANY NAME: _____

ADDRESS: _____

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.
FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
 PERMIT NO. 8241
 MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
 NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

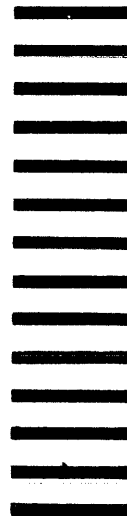
POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION

Publications and Graphics Division

215 Moffett Park Drive

Sunnyvale, California 94086



CUT ON THIS LINE

FOLD

FOLD

STAPLE

STAPLE