

**IBM System/3
Model 6
Conversational Utility Programs
Reference Manual**

Program Number 5703-UT1

Program Product

SC21-7528-4

Preface

This reference manual describes the capabilities of the conversational utility programs available for the IBM System/3 Model 6 user. The programs are:

- KSE — Keyboard Source Entry
- KDE — Keyboard Data Entry
- DIU — Data Interchange Utility

Related Publications are:

- *IBM System/3 Model 6 Operation Control Language and Disk Utility Programs Reference Manual*, GC21-7516
- *IBM System/3 Model 6 Operator's Guide*, GC21-7501

Fourth Edition (March 1973)

This is a major revision of, and obsoletes, SC21-7528-3. The following major changes have been made to the programs:

- Control statements (in card format) for KSE can be included in OCL procedures.
- Multiple KSE statements can be deleted with command key 05.
- Source modules can be included (command key 08) at any point in the current module.
- At end-of-job, the source module can be serialized to prevent the destruction of RPG tables.
- More than one source module can be created or maintained with one load of \$KSE.

This edition applies to version 08, modification 00 of the IBM System/3 Model 6 Conversational Utility Programs, program product number 5703-UT1, and to all subsequent versions and modifications until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the specifications herein; before using this publication in connection with the operation of IBM Systems, consult the latest IBM System/3 Newsletter Order Number GN20-2228 for the editions that are applicable and current.

Requests for copies for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901.

© Copyright International Business Machines Corporation 1971, 1972, 1973

CHAPTER 1. INTRODUCTION 1

Cataloged Control Statements 1

Keyboard Source Entry Program 1

Keyboard Data Entry Program 2

Format Descriptions 2

Data Interchange Utility Program 2

Command Keys 2

Function Keys 2

OCL Considerations 3

Inquiry Interrupt 3

CHAPTER 2. KEYBOARD SOURCE ENTRY (KSE) 4

Multiple KSE Jobs 4

Source Library Module 4

 KSE Statement Numbering 4

 Space Available in Source Module 5

 Source Library Module Naming 5

Format Descriptions 5

 Format Descriptions of KSE Source Statements 5

 Format Descriptions of KDE Data Records 7

 Format Description Command Key Restrictions 9

 Format Description End of Job Diagnostics 9

 Cancelling a Format Description Run 9

Source Statements 9

 KSE Formatted Source Statements 9

 KSE Free-Form Source Statements 11

 Source Statement Command Key Restrictions 11

KSE Command Keys 11

 Correcting Previous Statement or Inserting Statement 11

 Deleting Statements 12

 Including Another Module 12

 Skipping Rest of Statement 12

 KSE Command Key Usage 12

KSE Keyword-Response Summary Charts 18

 Creating or Maintaining Format Description of KSE Source Statements 18

 Creating or Maintaining Format Descriptions of KDE Data Records 20

 Creating or Maintaining Formatted KSE Source Statements 22

 Creating or Maintaining Free-Form KSE Source Statements 24

 Maintaining a Procedure 25

Including KSE Control Statements in Procedures 26

 Entering Control Statements 26

 Control Statement Diagnostics 26

 Control Statement Summary Chart 27

KSE Error Messages 28

OCL Load Cycle for KSE 32

KSE Sample Jobs 33

 KSE Sample Job 1. Create a Source Library Entry of KSE Format Descriptions 33

 KSE Sample Job 2. Create a Source Library Entry of Source Statements 34

 KSE Sample Job 3. Maintain an Existing Source Library Entry Containing KSE Format Descriptions 35

 KSE Sample Job 4. Maintain an Existing Source Library Entry Containing Source Statements 36

 KSE Sample Job 5. Maintain a Source Library Entry Containing KDE Format Descriptions 37

 KSE Sample Job 6. Building a Procedure to Create a New Source Module 38

 KSE Sample Job 7. Calling Procedure Built by Sample Job 6 39

 KSE Sample Job 8. Including Another Source Module and then Deleting Multiple Statements 40

 KSE Sample Job 9. Deleting Multiple Source Statements and then Including Another Source Module 42

 KSE Sample Job 10. Correcting Statements in Two Source Modules in one KSE Run 44

 KSE Sample Job 11. Maintaining One Source Module and Creating Another in One KSE Run Using Cataloged KSE Control Statements 45

 KSE Sample Job 12. Calling a Procedure Containing Errors in KSE Control Statements 46

CHAPTER 3. KEYBOARD DATA ENTRY (KDE) 47

KDE Data Files 47

 Key Fields 47

KDE Record Format Descriptions 47

KDE Initialization Keyword-Response Sequence 48

 KDE Keyword-Response Summary Chart 49

Entering Data Records 50

 Alphameric Fields 50

 Signed Numeric Fields 50

 Unsigned Numeric Fields 50

 Automatic Duplicate Fields 50

 Skip Fields 50

 Packed Fields 50

 Self-Check Fields 50

 Free-Form Records 51

KDE Accumulators 52

 Batch Total Accumulators 52

 Final Total Accumulators 52

 Updating Accumulators 52

KDE Command Keys 53

 Correcting Current Record 53

 Correcting Previous Record 53

 Deleting Previous Record 54

 Skipping Rest of Record 54

 KDE Command Key Usage 54

Including KDE Control Statements in Procedures 59

 Entering Control Statements 59

 Control Statement Diagnostics 60

 Control Statement Summary Chart 61

OCL Load Cycle for KDE 62

KDE Error Messages 63

KDE Sample Jobs 66

 KDE Sample Job 1. Creating a New Keyboard Data Entry File 66

 KDE Sample Job 2. Maintaining a Keyboard Data Entry File 68

 KDE Sample Job 3. Building a KDE Procedure to Create a New File 70

 KDE Sample Job 4. Calling Procedure Built by Sample Job 3 71

KDE Sample Job 5. Calling a KDE Procedure to Maintain the File Built by Sample Job 4	72	Converting from System/3 BASIC to Disk System Management	78
KDE Sample Job 6. Calling a Procedure With Delayed Responses	73	Considerations for Converting Numeric Fields from System/3 BASIC	79
KDE Sample Job 7. Calling a Procedure to Create a Free-Form File	74	Definitions That Cause Most Significant Digit to be Dropped	79
KDE Sample Job 8. Calling a Procedure Containing Errors in KDE Control Statements	75	DIU Command Key Chart	79
CHAPTER 4. DATA INTERCHANGE UTILITY (DIU)	76	Methods of Supplying OCL and DIU Statements Using the Keyboard to Supply the DIU Control Statements	80
Data File Conversion	76	DIU Keyword Response Summary Chart	80
System/3 BASIC Data Fields	77	DIU Control Statements on Cards	82
Disk System Management Data Fields	77	DIU Control Card Format and Sequence	82
Describing Disk System Management Fields	77	DIU Control Statements Included in Procedure	82
Disk System Management Field Description Diagnostics	78	OCL Load Cycle for DIU	83
Considerations for Converting Alphameric	78	DIU Error Messages	84
Converting from Disk System Management to System/3 BASIC	78	DIU Sample Job	88
		INDEX	89

How This Manual Is Organized

This manual is divided into four chapters. Chapter 1 is an introduction to the Conversational Utilities and the Keyword-Response sequence. Chapters 2, 3, and 4 describe KSE, KDE, and DIU respectively. Charts within each chapter summarize the steps an operator should take to use the conversational utility programs.

The IBM System/3 Model 6 conversational utility programs are:

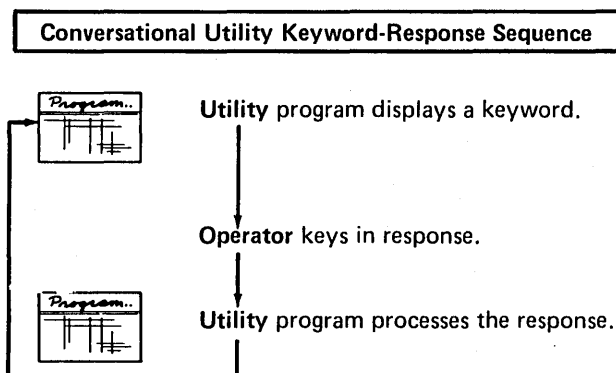
- KSE – Keyboard Source Entry
- KDE – Keyboard Data Entry
- DIU – Data Interchange Utility

The conversational utility programs converse with the operator by means of printed messages and questions. The questions are called *keywords*. The operator answers each keyword using the keyboard. The keyed answers are called *responses*. Responses tell the utility program what kind of data is to be processed, where it is located, and what type of processing is to be done.

A keyword and its corresponding response are called a *control statement*. The series of control statements at the start of each conversational utility run is called the *initialization keyword-response sequence*. This sequence is used to identify:

- Type of run.
- Type of statement to be processed.
- Name and unit of source library module or data file to be processed.

A keyword-response sequence operates in the following manner:



After the conversational utility keywords are printed, the print carriage is repositioned for the operator's response. The operator types in the response, which is printed on the same print line as the keyword. The operator presses the program start key to indicate the end of the response. For a discussion of the program start key and other function keys, see *Function Keys* later in this chapter.

After the program start key is pressed, the conversational utility processes the response. If the response is valid, the next keyword is printed. If an invalid response is given, an error message is printed and the same keyword is printed again.

By printing a keyword, the conversational utilities *prompt* the operator for a relevant response. In this manual, printing a keyword is referred to as prompting.

Note: Responses must be keyed in exactly as shown in the keyword-response summary charts in Chapters 2 through 4. If a blank (space) is entered before a response, the response will be processed as if only the program start key was pressed.

CATALOGED CONTROL STATEMENTS

The control statements (keywords with responses) for the conversational utility programs can be included in an OCL procedure. Chapters 2-4 describe cataloged procedures for KSE, KDE, and DIU, respectively.

KEYBOARD SOURCE ENTRY PROGRAM

The Keyboard Source Entry (KSE) conversational utility program is used to create and maintain source library modules. Input to KSE is entered from the keyboard. The output is written into a source library module on disk. The input data is also printed to serve as a record of what was entered.

The KSE program has two run sequences to perform the following functions:

1. Create a source library module.
2. Maintain an existing source library module.

KEYBOARD DATA ENTRY PROGRAM

The Keyboard Data Entry (KDE) conversational utility program processes data files. The operator types the data records on the keyboard; the KDE program processes the records and writes the data into a disk file. The input data is also printed to serve as a record of what was entered.

The KDE program has two run sequences to perform the following functions:

1. Create a disk data file.
2. Maintain an existing disk data file.

FORMAT DESCRIPTIONS

The KSE and KDE utility programs enable the operator to enter data from the keyboard. To aid the operator in entering the data correctly and easily, *format descriptions* are used as an input control program. A format description consists of a header statement plus a field description statement for each field in the record being described. The field descriptions define the type of data in each field and the field length. The total of the field lengths in a format description must equal the record length in the header statement.

Because format descriptions reside in the source library, both format descriptions of KSE source statements and format descriptions of KDE data records are created and maintained by KSE. (See *Chapter 2. Keyboard Source Entry; Format Descriptions of KDE Data Records* and *Format Descriptions of KSE Source Statements*.)

DATA INTERCHANGE UTILITY PROGRAM

The Data Interchange Utility (DIU) program changes the format of System/3 BASIC data files so they can be used with the disk system management programs. Data files can also be converted to System/3 BASIC format. This program, therefore allows programmers at installations with System/3 Model 6 co-resident programming systems to convert data files created on one system to the format required to run on the other.

COMMAND KEYS

The System/3 Model 6 keyboard contains a group of keys to the left of the main keyboard. These keys, called command keys, are used during the operation of the conversational utility programs to control the functions being performed by the program.

Command key uses differ among KSE, KDE, and DIU. The command key charts in Chapters 2 through 4 list the uses of the command keys. Whenever the keyboard is ready for an operator response, the field lights indicate which command keys may be used. If the field light is on, the command key may be used. An exception to this is field light 03, which indicates the condition of the automatic duplication indicator in KSE and KDE. If the automatic duplication indicator is on, field light 03 is on.

FUNCTION KEYS

There are nine function keys on the keyboard that are used when entering data into the system. These keys are:

Program Start (PROG START) indicates end of field. The printer positions itself at the start of the next field.

Field Erase (ERASE) erases all the data in a field so the operator can retype the entire field. This must be done before the program start, enter plus, or enter minus key has been used to define the end of the field.

Backspace (BKSP) backspaces over one or more characters in a field so the operator can retype the characters. This must be done before the program start, enter plus, or enter minus key has been used to define the end of the field.

Space places blanks in alphameric fields.

Enter Minus (ENTER-) assigns a minus sign to a numeric field and signals the end of that field. It must not be used until all the numeric data for the field has been entered.

Enter Plus (ENTER+) indicates end of field. The printer positions itself at the start of the next field.

Tab (TAB) duplicates alphameric fields (types A, D, C, and T) one character at a time. The characters are duplicated as long as the key is depressed. When the key is held depressed, duplication stops at the end of the field and the program start key must be pressed. To duplicate the next field (if alphabetic) the tab key must be pressed again.

Return (RETURN) causes one line space. The printer does not return to position one.

Form Skip (↑) causes the printer to eject the paper to the start of the next form.

OCL CONSIDERATIONS

The KSE, KDE, and DIU conversational utility programs can be loaded using either the Operation Control Language (OCL) LOAD or CALL cycles. The CALL cycle requires the prior creation of a procedure by using the OCL BUILD cycle.

The control statements for KDE and DIU can be included in the OCL procedures rather than supplied through the keyword-response sequence.

INQUIRY INTERRUPT

The inquiry request switch can be used to interrupt KDE. KSE and DIU cannot be interrupted by an inquiry request.

Chapter 2. Keyboard Source Entry (KSE)

The Keyboard Source Entry utility program creates and maintains source library modules using data the operator types on the keyboard. KSE processes the input data and writes it as a permanent entry in the source library of the disk specified by the operator.

You can use the KSE utility to either create or maintain source library modules.

Operations performed by KSE are controlled by the operator's responses during the initialization keyword-response sequence.

The following functions can be performed by KSE:

- Create or maintain a source library module of format descriptions of source statements.
- Create or maintain a source library module of format descriptions of KDE records.
- Create or maintain a source library module of source statements.
- Create or maintain a procedure in the source library.

Note: KSE uses available space on disk for a work area. Because of this work area, scratch files are destroyed. KSE finds the largest work area available on all disk drives that are powered up. This work area is allocated to handle the largest number of source statements, not for best performance. As a result, performance may vary from run to run depending on the location of the available work area.

Multiple KSE Jobs

More than one source library module can be maintained with one KSE load. When work on one source library module is complete, the operator presses command key 06, causing KSE to prompt END-OF-JOB. The operator can respond with YES,START if work is to be done on another module. After KSE writes the first module into the source library, it prompts FORMAT DESCRIPTION for another job. If cataloged control statements were used for the first job, the responses from the control statements are used again.

SOURCE LIBRARY MODULE

A source library module is a group of 96-character statements with a module name. Source library modules can contain:

- Source Statements – RPG source statements, sort specifications, etc.
- Procedures – Built by OCL BUILD cycle, KSE, or Library Maintenance program.
- Format descriptions – Of either KDE records or KSE statements.

KSE Statement Numbering

The KSE conversational utility program assigns a 5-digit statement number to each keyed input statement in a source library module. Statement numbering is performed when modules are created. Statement numbers are printed in the first five positions of the print line. The KSE program increments the statement numbers by ten: 00000, 00010, 00020, and so on. This permits the operator to insert up to nine new statements between two previously keyed statements.

When a source module created or maintained by the KSE utility program is placed in the source library at end-of-job, the statements are resequenced by 10 to allow for any statements inserted or deleted. The operator can specify that the statement numbers be inserted into the statements. A YES,SERIAL or YES,RPG response to the END-OF-JOB prompt will cause SERIAL START POSITION to be prompted. The operator must then tell KSE where to put the statement number in the source statement by entering the starting position (1 through 92 are valid entries) or by pressing the program start key in response to SERIAL START POSITION. Pressing the program start key will cause positions 1-5 to be used. Any data already in these positions will be overlaid by the statement number.

If YES,SERIAL is specified, all statements in the module will have statement numbers inserted. If YES,RPG is entered, the statement numbers are inserted until a statement with **b in positions 1-3 is read. This prevents RPG II tables from being destroyed. YES,SERIAL and YES,RPG can be combined with YES,START at end of job. For example; YES,SERIAL,START causes KSE to serialize and restart.

When the module is called for maintenance, statement numbers are again assigned, whether or not they are included in the source statements.

For example, the following statements are read from a source library module for maintenance and assigned statement numbers:

00000	ALICE
00010	BARB
00020	CLAIR
00030	DEBBIE
00040	EARL
00050	FRED
00060	GEORGE

The operator adds the following statements:

00021	DALE
00031	DIANE
00032	DICK
00033	DON

When the module is next called for maintenance, the statements would be assigned statement numbers as follows:

00000	ALICE
00010	BARB
00020	CLAIR
00030	DALE
00040	DEBBIE
00050	DIANE
00060	DICK
00070	DON
00080	EARL
00090	FRED
00100	GEORGE

Space Available in Source Module

The number of source statements or format descriptions that can be added to a source module depends on the size of the module and the format of the statements. KSE will print the message:

nnnnn NEW STATEMENTS MAY BE ADDED
TO THE SOURCE MODULE

where nnnnn is the largest number of new statements that can be added without danger of exceeding the source library size. After nnnnn statements have been entered, the KSE run should be ended. If more statements must be added, KSE should be loaded again for a maintenance run. KSE will then print the number of statements that can safely be added.

Source Library Module Naming

Names used for KSE source library modules can be from one to six characters long. Any valid System/3 characters can be used in a name, except commas, blanks, quotes (apostrophes), and periods. The first character must be alphabetic (A-Z, \$, #, or @). These names are used to identify the source programs or procedures in the source library. DIR, SYSTEM, and ALL are names reserved for system use and cannot be used.

FORMAT DESCRIPTIONS

The KSE utility program is used to create and maintain format descriptions for both KSE source statements and KDE data records.

Format Descriptions of KSE Source Statements

The fields in a KSE formatted source statement must be defined by field description statements before the source statement can be processed. The group of field description statements required to define all the fields in a source statement is called a format description. All the format descriptions required to run a KSE job can be placed in a module in the source library. The KSE program is used to create and maintain the KSE format description modules in the source library.

Format descriptions for KSE are created in the following manner. (See *KSE Keyword-Response Summary Charts; Creating or Maintaining Format Descriptions of KSE Source Statements*.) The operator's YES response to the prompt **FORMAT DESCRIPTION** tells the KSE program that this run will be used for processing field description statements. The operator's KSE response to the prompt **FORMAT TYPE** tells KSE that the statements to be keyed in during this run will be KSE field descriptions.

After prompting for type of run (**NEW SOURCE MODULE**) and the name and unit of the source module, KSE prints a header statement identifying the format description that will contain your KSE field descriptions.

KSE prints the header statement in the following format:

xxxxx Hdd96

Where:

xxxxx is the statement number

H indicates that this statement is a format header.

dd is the format description number. The format description number identifies the entire group of field description statements and must be used as the response to the prompt **SELECT FORMAT NUMBER** when you are working with the KSE source statements that use this format.

96 is the length of KSE source statements.

Following the header line, KSE prompts statement numbers. The operator then responds with the field description statements as follows:

xxxxx tll

Where:

xxxxx is the statement number

t is the field type entered by the operator. Valid field types are:

- N – This field will contain only signed numeric characters. The characters are right-adjusted in the field with zeros filled to the left.
- R – This field will contain only unsigned numeric characters. The characters are right-adjusted in the field with blanks filled to the left.
- A – This field will contain any valid System/3 character. The characters are left-adjusted in the field with blanks filled to the right.
- D – Contents of this field can be automatically duplicated in each source statement (see *KSE Command Key Chart; Command Key 3*)
- S – This field will be automatically filled with blanks in each KSE source statement.

ll is the field length entered by the operator. Field types N and R can be up to 15 characters long. A, D, and S fields can be 1 to 96 characters long.

Field description statements identifying the fields must be keyed in the same order as the fields will appear in the source statements. When field descriptions for all 96 positions have been entered, the format description for that source statement is completed. KSE then prints the header statement for the next format.

KSE Format Description Diagnostics

While the operator is keying in the field description statements, the KSE utility diagnoses them for errors. If an error is found, an error message is displayed on the same line as the invalid field description statement. The KSE utility then repositions the printer carriage and allows the operator to rekey the field description statement.

The error checking routines in the KSE utility program diagnose the operator keyed field description statements for the following errors:

1. Field type invalid. Valid field types are: N (signed numeric), R (unsigned numeric), A (alpha-numeric), D (automatic duplicate), and S (skip).
2. Field length invalid. The length for numeric fields (R and N) cannot be greater than 15; the length for A, S, and D fields cannot be greater than 96.
3. Statement length invalid. Length given for the current field makes total length of all fields greater than 96.

Format Descriptions of KDE Data Records

Before a data record can be processed by KDE using format control, the fields of the record must be defined by the field description statements. The group of field description statements required to define all the fields in a data record is called a *format description*. All the format descriptions required to run a KDE job can be placed in a module in the source library. The KSE program is used to create and maintain the KDE data record format description modules in the source library.

Format descriptions for KDE are created in the following manner (see *KSE Keyword-Response Summary Charts: Creating Format Descriptions of KDE Data Records* in this chapter).

The operator's YES response to the prompt **FORMAT DESCRIPTION** tells the KSE program that this run will be used for processing field description statements. The operator's KDE response to the prompt **FORMAT TYPE** tells KSE that the statements to be keyed in during this run will be KDE field descriptions.

After determining type of run (new or maintenance) and the name and unit of the source module, KSE prints a header statement identifying the format description that will contain your KDE field descriptions.

KSE prints the header statement in the following format:

```
xxxxx Hdd nnn or nnn+nnn
```

Where:

xxxxx is the statement number

H indicates that this statement is a format header.

dd is the format description number. The format description number identifies the entire group of field description statements and must be used as the response to the prompt **SELECT FORMAT NUMBER** when you are working with the KDE data records that use this format.

nnn or nnn+nnn is the record length provided by the operator. KDE data records may be from 1 to 256 characters long. The operator must key in the record length as a response to the header line prompt.

- nnn indicates that the record is to be printed on one print line.

If the record length, number of fields, and type of fields prevent the entire record from printing on one line, KDE will divide the record into more print lines.

- nnn+nnn indicates that the record is to be printed on two print lines.

All records in a KDE data file must be the same length.

In creating format descriptions of KDE data records, the operator must keep in mind that every field in a KDE record, including the last field, is followed by a blank when printed. Therefore, the number of fields per record is limited to the page width minus the line length if the record is split into two lines.

Example: page width = 132, record length = 120.
The number of fields is limited to 12.

In determining record length and page width, it must be remembered that packed numeric fields require twice the number of print positions as record positions.

Following the header line, KSE prompts field description lines as follows:

xxxxx tIII [AN]

Where:

xxxxx is the statement number supplied by KSE

t is the field type entered by the operator.
Valid field types are:

- N – This field will contain only signed numeric characters. The characters are right-adjusted in the field with zeros filled to the left.
- R – This field will contain only unsigned numeric characters. The characters are right-adjusted in the field with blanks filled to the left.
- A – This field will contain any valid System/3 character. The characters are left-adjusted in the field with blanks filled to the right.
- D – The contents of this field can be automatically duplicated in each KDE data record (see *KSE Command Key Chart; Command Key 3* in Chapter 2).

- S – This field will be automatically filled with blanks in each KDE data record.
- C – This is a modulus 11 self-check field.
- T – This is a modulus 10 self-check field.
- P – Packed numeric. The numeric data will be packed two characters to a byte before being placed in the record.

III is field length entered by the operator. P fields can be up to 8 characters long. C, T, R, and N fields can be from 1 to 15 characters long. A, D, and S fields can be from 1 character to the length of the print line.

[An] is an accumulator reference. The brackets indicate that this entry is optional. Brackets are not keyed as part of the response. (For further information, see *Chapter 3. KDE Accumulators.*)

Field description statements identifying the field must be keyed in the same order as the fields will appear in the data records. When field descriptions for the entire record have been entered, the format description for that data record is completed.

If KDE is to provide the key field for the records being supplied, the first field description in each record must be an N or A type with a length of five (or P003 if a packed numeric key field is desired).

KDE Format Description Diagnostics

While the operator is keying in the field description statements, the KSE utility diagnoses them for errors. If an error is found, an error message is displayed on the same line as the invalid field description statement. The KSE utility then repositions the printer carriage and allows the operator to rekey the field description statement.

The error checking routines in the KSE utility program diagnose the operator keyed field description statements for the following errors:

1. Record length in header statement greater than 256.
2. Field type invalid. Valid field types are: N (signed numeric), R (unsigned numeric), A (alphameric), D (automatic duplicate), C and T (self-check), S (skip), and P (packed).
3. Length of a C, T, R, or N field greater than 15.
4. Length of P field greater than 8.
5. Invalid accumulator number requested.
6. Invalid field assigned to an accumulator.
7. Record length in current header statement not equal to record length in previous header statements for this file.
8. Length of current field causes total length of the record to be greater than length specified in header statement.
9. Line length in header statement greater than 132.
10. Field length causes the field to be split between two print lines.

If an error is found in a field description statement, KSE prints an error message and reprompts the statement number of the invalid field description. The operator may then rekey the field description.

Format Description Command Key Restrictions

During a format description run, certain command key functions are restricted. Command keys 02, 03, 04, and 07 are not allowed by the program. Command keys 01, 05, 06, and 08 are allowed only as a response to a header statement. This restriction ensures that an entire source statement or data record is described.

Format Description End of Job Diagnostics

The KSE end of job processing for a format description run displays and error checks all the statements in the source library module if command keys 01, 05, or 08 were used during the run. If any format is found to contain an error, the error must be corrected before the format is put into the source library module. Rather than correct the error, the operator can enter CANCEL. This causes all formats listed before the error format to be put into the module. The error format and all formats after it are dropped.

Cancelling a Format Description Run

During a format description run the operator may enter CANCEL instead of a field description to cancel the job. KSE will immediately go to end of job. Any processing done during the run will be ignored.

SOURCE STATEMENTS

There are two types of source statements in the source library, formatted and free-form. Formatted source statements have each field in the 96-character statement described by a field description. Free-form source statements are processed as 96-character statements with just one 96-character alphameric field.

Both formatted and free-form statements can be in the same library module.

KSE Formatted Source Statements

A KSE formatted source statement run is used to place source library modules containing source statements on disk. The operator tells the KSE utility that this run will be for processing KSE formatted source statements by responding NO to the prompt FORMAT DESCRIPTION. KSE then prompts FORMAT NAME and FORMAT UNIT for the source library module containing the format descriptions of the source statements to be processed. The operator has an option at this time to have KSE display the format descriptions. A YES response to DISPLAY FORMATS causes KSE to display the format descriptions; a NO response bypasses the display. All format descriptions are checked for validity.

The operator then indicates whether this run is to create a source library module or maintain an existing source library module by answering either YES or NO to NEW SOURCE MODULE.

KSE then prompts the operator for the name and disk unit of the source library module to be processed, and prints the approximate number of new statements the module may accept during this run.

The operator then gives KSE the number of the appropriate format description, and KSE prints an asterisk (*) in the starting print position of each field described.

If the current run is a maintenance run, KSE assigns statement numbers to the statements already in the module and prints the next sequential statement number for statements to be added. The print head is then positioned for entering the first field. If the current run is a source library module creation run, KSE displays the statement number 00000, and positions the print head at the start location of the first field.

The operator may then begin keying in source statements or use the command keys for processing. The operator keys in all the data for a field and then presses the program start key to indicate the end of data for the field.

Alphameric Fields

Alphameric (A-type) fields are printed as the characters are entered. If the operator presses the program start key (indicating end of field) before entering the entire field, the remaining positions are filled with blanks. Alphameric fields (A and D types) can be duplicated one character at a time by using the tab key. As long as the tab key is held depressed, characters in the field are duplicated from the previous statement. At the end of the field, the program waits until the tab key is released. The program start key must then be pressed to enter the field into the statement.

Signed Numeric Fields

Signed numeric (N-type) fields are not printed as they are entered. Because signed numeric fields are right-justified and the function key (enter plus or enter minus) determines the sign, the field is printed after the end of the field is indicated. An extra print position is required to print the sign after the units position of the field. To zero-fill a signed numeric field, the operator must enter at least one zero. If no data is entered (only the program start key is pressed), the field is blank-filled except for a zero in the low order position.

Unsigned Numeric Fields

Unsigned numeric (R-type) fields are not printed as they are entered. Because they are right justified, the characters are not printed until the end of the field is indicated. The leading positions are blank-filled. Because a sign is not printed, an extra print position is not required as it is for signed numeric (N-type) fields. The program start key or the enter plus key can be used to enter the data.

Automatic Duplicate Fields

D-type fields are automatically duplicated from the previous statement if the automatic duplication indicator is on. Field light 03 indicates the condition of the automatic duplication indicator.

When a different format is selected, automatic duplication is suppressed for the first statement of the new format. However, field light 03 still shows the condition of the automatic duplication indicator.

Skip Fields

S-type fields are filled with blanks automatically.

KSE Free-Form Source Statements

The KSE utility can create and maintain source library statements which are not subject to format control. This type of statement is known as free-form. No previous KSE run is required to create format descriptions when processing free-form source statements. A free-form statement is processed as having one alphanumeric field (A) 96 characters long.

The operator tells the KSE utility program that the current run is for processing free-form source statements by using one of the following methods:

1. Pressing the program start key as the response to **FORMAT DESCRIPTION**.
2. Responding **YES** to the prompt **FORMAT DESCRIPTION** and then pressing the program start key as the response to **FORMAT NAME**.
3. Responding **NO** to the prompt **FORMAT DESCRIPTION** and then pressing the program start key as the response to **FORMAT NAME**.

The remainder of the keyword-response sequence for processing free-form source statements is consistent with the keyword-response sequence for processing formatted source statements. However, since formats are not used with free-form statements, **FORMAT NAME**, **FORMAT UNIT**, and **DISPLAY FORMATS** are not prompted; field starting print positions are not displayed.

The operator can also enter free-form source statements during a formatted source statement run by pressing command key **02** and responding **00** to the prompt **SELECT FORMAT NUMBER**. Field starting print positions are not displayed.

The free-form processing function of KSE should not be used to process **RPG II** source statements.

Source Statement Command Key Restrictions

When the operator is entering formatted source statements, command keys **01**, **02**, **05**, **06**, and **08** are only allowed before the first field of a source statement is entered. This ensures that an entire 96 character statement is entered before the operator changes functions.

During a free-form run, command key **03** and command key **02** are allowed only if free-form was entered from a formatted run by responding **00** to the prompt **SELECT FORMAT NUMBER**.

KSE COMMAND KEYS

KSE command keys perform the following functions:

<i>Command Key</i>	<i>Function</i>
01	Insert a statement or correct a previous statement.
02	Select a different format description.
03	Change the contents of an automatic duplication field. The condition of the automatic duplication indicator and of field light 03 is reversed.
04	Duplicate a field.
05	Delete a statement or statements.
06	End the job.
07	Duplicate the remainder of this statement.
08	Include another source module in this module.
04 & 07	Skip the remainder of this statement.

The field lights are used in conjunction with command key operations. Field light 03 indicates the condition of the automatic duplication indicator. If field light 03 is on, the indicator is on. The other field lights indicate when command keys can be used. When a particular field light is on, its corresponding command key can be used.

Correcting Previous Statement or Inserting Statement

The operator can correct a previously entered statement or insert a statement between two previously entered statements by pressing command key **01**. The KSE program prompts **CORRECTING STATEMENT NUMBER**, and the operator enters the statement number. If the statement number matches the statement number of a previously entered statement, KSE displays that statement as it currently appears. The operator then enters the corrected statement. The statement being corrected is also used as the previous statement for duplication. Thus, any fields that are duplicated will be unchanged. If the statement number entered by the operator does not match a statement number in the module, the new statement is inserted in the module.

Deleting Statements

KSE statements can be deleted by using command key 05. The KSE program prompts DELETING STATEMENT NUMBER. To delete only one statement, the operator enters the statement number of that statement.

The operator can delete a block of contiguous statements by pressing the program start key as the response to the prompt DELETING STATEMENT NUMBER. KSE then prompts FROM-. The operator enters the statement number of the first statement to be deleted. KSE then prompts TO-, and the operator enters the statement number of the last statement to be deleted. KSE then deletes all statements within these limits. The FROM statement number must be a valid statement number and must be smaller than or equal to the TO statement number. The TO statement number need not be a valid number. For example; The operator can delete all statements in a module from statement number 08326 until the end of the module by entering FROM-08326 to TO-99999.

Including Another Module

When creating or maintaining a source library module (either source statements or format descriptions), the contents of another source module can be included. When the operator presses command key 08, KSE prompts for the name and unit of the source library module to be included, and then prompts INCLUDE AFTER STATEMENT NUMBER-. If the operator presses the program start key as a response, the module is added after the last statement in the present module. If the operator enters a valid statement number, KSE inserts the statements of the included module after that statement.

The added statements are assigned statement numbers and printed as they are included. After the new statements are included, all statements in the original module after the point of inclusion are resequenced by ten. Therefore, if including statements at several points in the original module, the operator should begin including at the end of the module (highest statement number) and work towards the beginning of the module.

Skipping Rest of Statement

The operator can skip the remaining fields of a statement by pressing command keys 04 and 07. The remaining fields are filled as follows:

A (Alphameric) – blanks

N (Signed numeric) – zeros

R (Unsigned numeric) – blanks

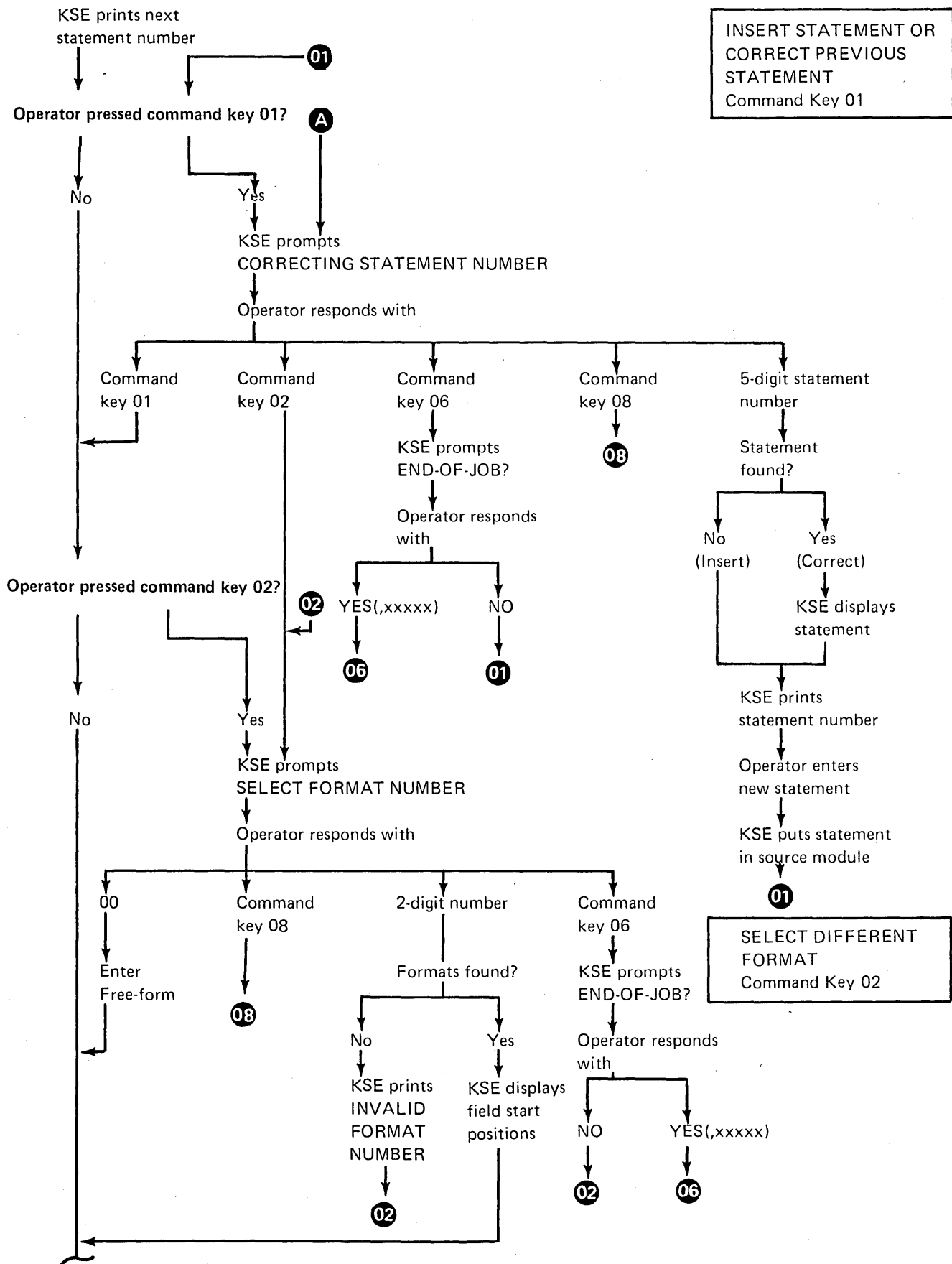
D (Automatic Duplicate) – blanks if automatic duplication indicator (field light 03) is off.

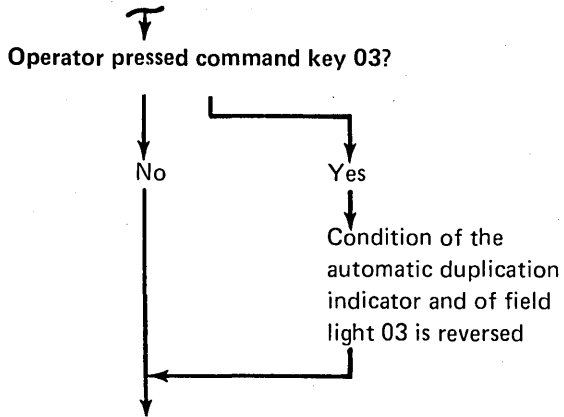
D (Automatic Duplicate) – duplicated from previous statement if automatic duplication indicator (field light 03) is on.

S (Skip) – blanks

KSE Command Key Usage

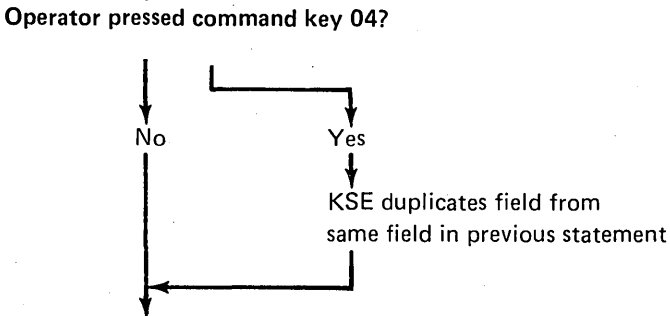
The following chart shows the operator-system interaction when the operator is using command keys.



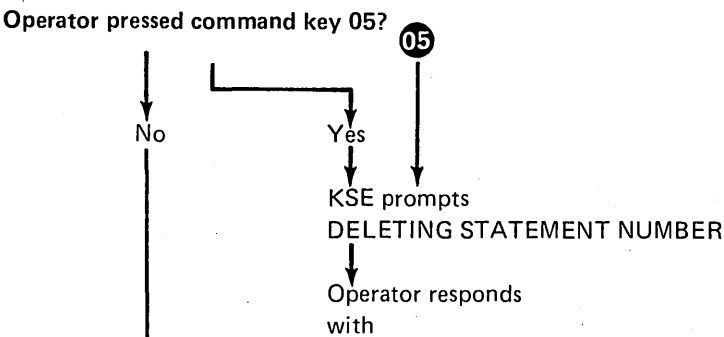


CHANGE CONTENTS OF
AUTOMATIC DUPLICATE
FIELD
Command Key 03

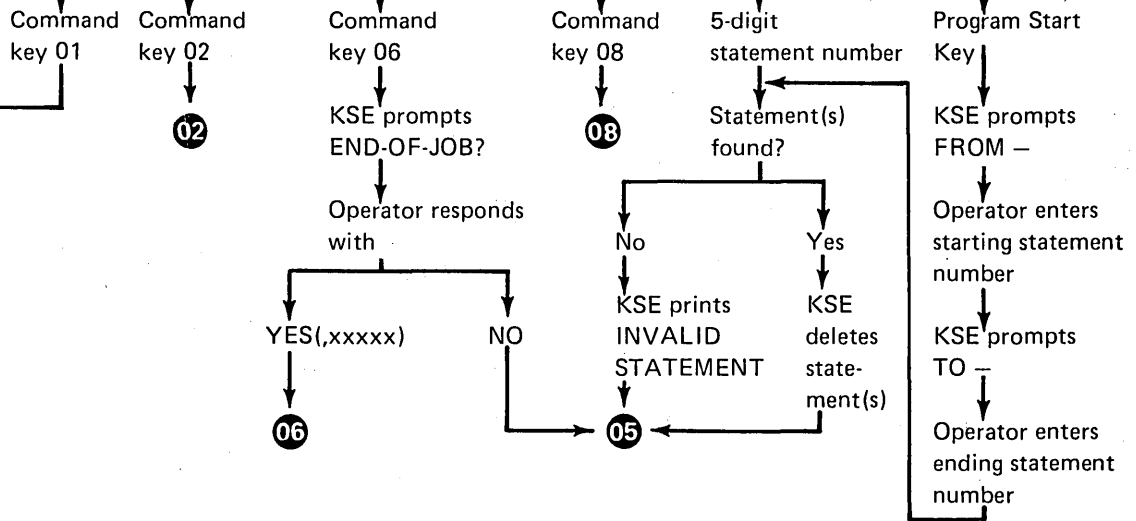
Note: If automatic duplication indicator is on when operator changes formats, automatic duplication is suppressed for the first record processed under the new format. Automatic duplication will occur for all succeeding records.



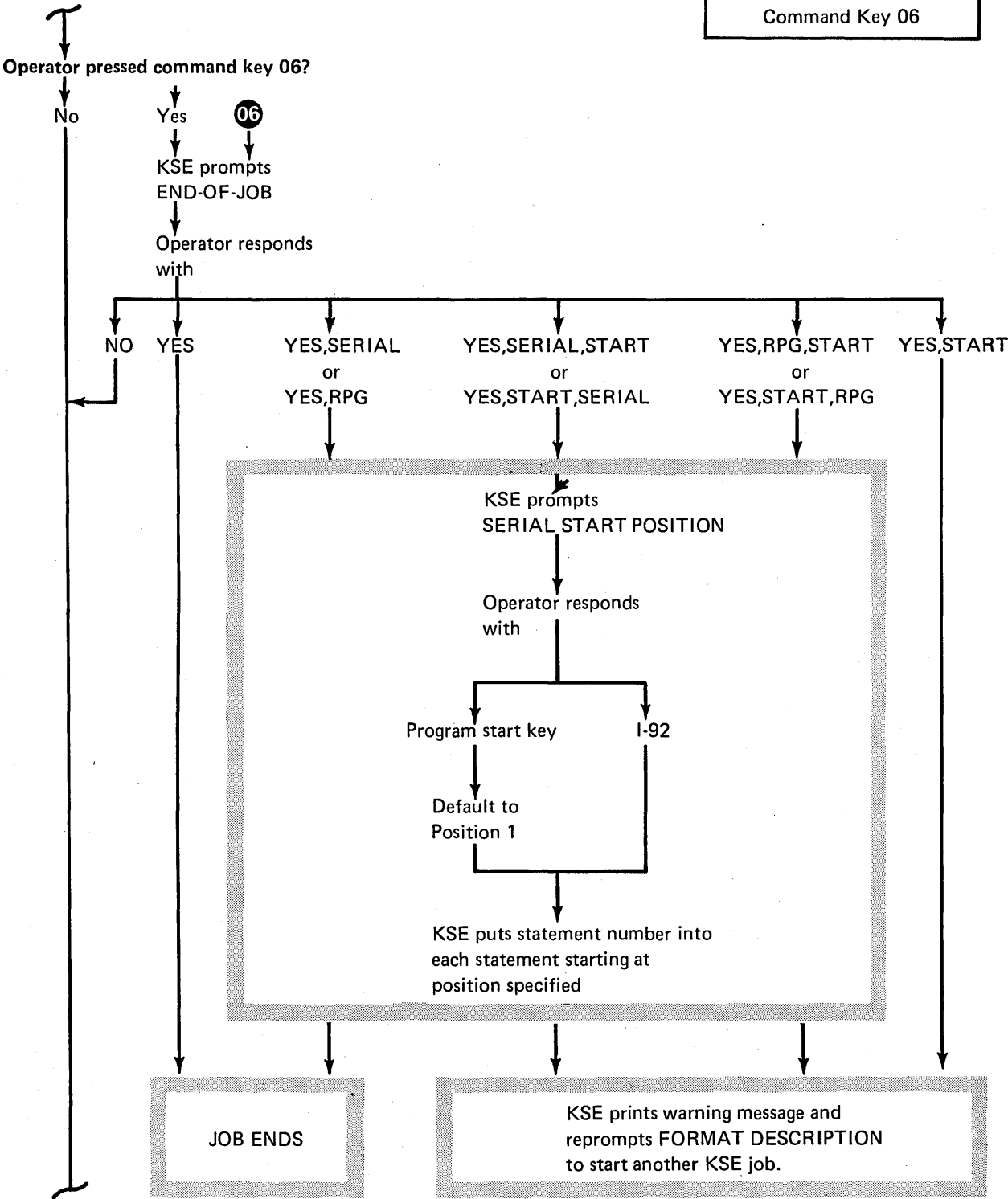
DUPLICATE FIELDS
Command Key 04

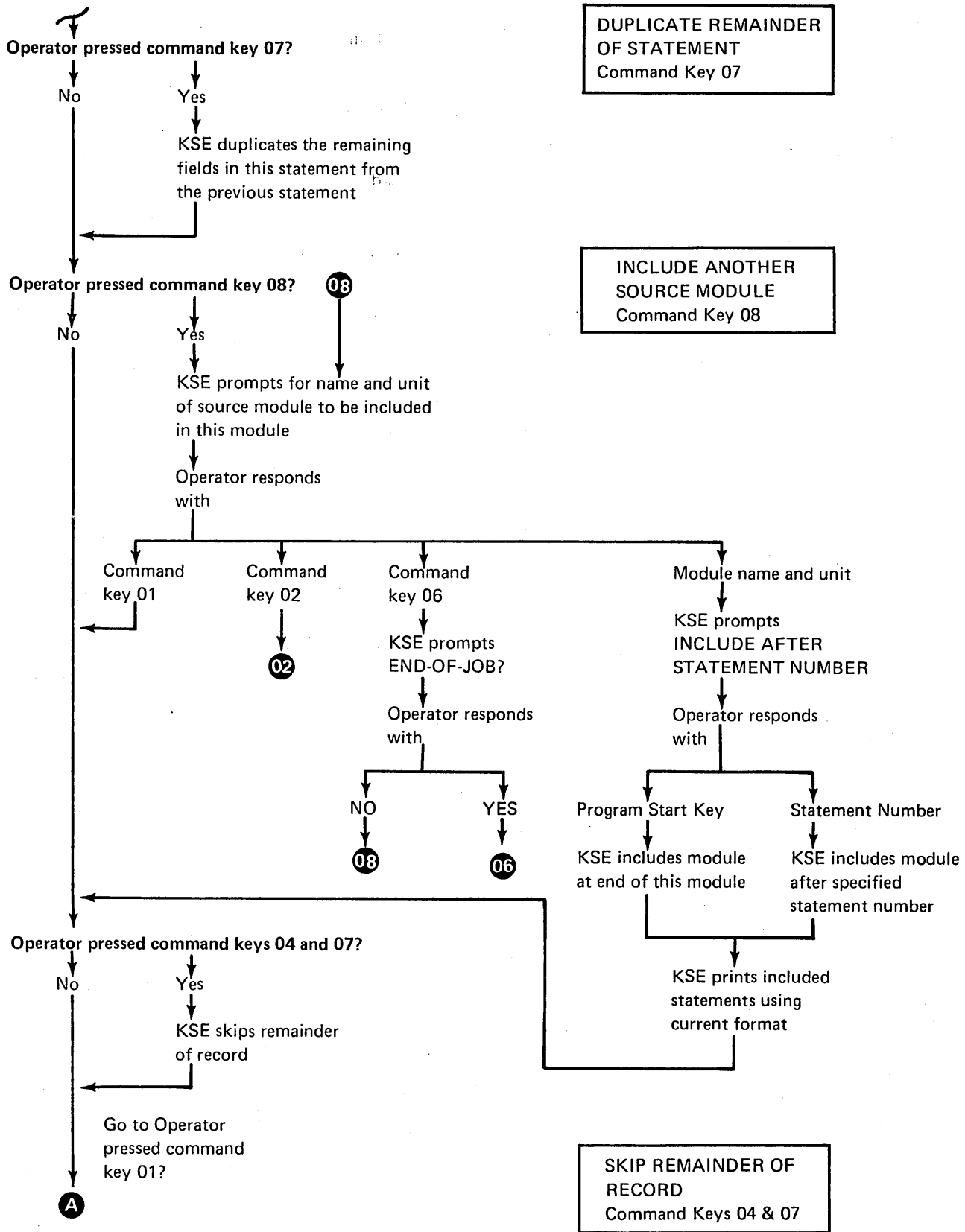


DELETE STATEMENT
Command Key 05



END THE JOB
Command Key 06





KSE KEYWORD-RESPONSE SUMMARY CHARTS

The following charts show the sequence of control statements for each of the functions of KSE. The KSE program diagnoses the YES, NO type responses. If a response is given that is not shown in the chart, KSE reprompts the same keyword for a valid response.

Creating or Maintaining Format Description of KSE Source Statements

Keyword	Response	Considerations
FORMAT DESCRIPTION	YES	You are processing format descriptions.
FORMAT TYPE	KSE	You are processing KSE format descriptions.
NEW SOURCE MODULE	YES	You are creating a new source module containing KSE format descriptions.
	NO	You want to maintain the statements in an existing source library module containing KSE format descriptions.
SOURCE MODULE NAME	1-6 character name	You supply the name of the source library module you want to create or maintain.
SOURCE MODULE UNIT	F1, F2, R1, or R2	You supply the unit of the disk on which the source library module you want to create or maintain is located.

WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.

nnnn NEW SOURCE STATEMENTS MAY BE ADDED TO THE SOURCE MODULE

00000 H0196 (create only)	Program start key	KSE supplies the first header
00010 (create only)	tll	You provide the type and length for the first field description.
000n0 (create only)	tll	KSE continues to prompt you for more field descriptions until you have accounted for 96 positions.
nnnn Hdd96	Program start key	KSE provides the next header when you complete 96 characters of description. If you press the program start key, KSE prompts you for the next field description.
	Command key 01	KSE prompts correction sequence
	Command key 05	KSE prompts for deletion
	Command key 06	KSE prompts END-OF-JOB
	Command key 08	KSE prompts for included module

Creating or Maintaining Format Description of KSE Source Statements (continued)

Keyword	Response	Considerations
nnnnn	tll	You pressed the program start key. You should now supply the type and length of a field. KSE continues to prompt for additional field descriptions until you have accounted for 96 characters. KSE then displays the next header line.
END-OF-JOB	YES or YES,START	If command keys 01, 05, or 08 were used, KSE displays and checks entire source library module for errors. If errors are found, KSE prompts the statement number so operator can correct the statement. If YES,START is entered, KSE prompts FORMAT DESCRIPTION for another job.
	NO	KSE prompts the next statement number and header.

317-104-0

Creating or Maintaining Format Descriptions of KDE Data Records

Keyword	Response	Considerations
FORMAT DESCRIPTION	YES	You are processing format descriptions.
FORMAT TYPE	KDE	You are processing KDE format descriptions.
NEW SOURCE MODULE	YES	You are creating a new source module containing KDE format descriptions.
	NO	You want to maintain the statements in an existing source library module containing KDE format descriptions.
SOURCE MODULE NAME	1-6 character name	You supply the name of the source library module you want to create or maintain.
SOURCE MODULE UNIT	F1, F2, R1, or R2	You supply the unit of the disk on which the source library module you want to create or maintain is located.
WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.		
nnnnn NEW SOURCE STATEMENTS MAY BE ADDED TO THE SOURCE MODULE		
00000 H01 (create only)	nnn or nnn+nnn	You supply the record length to be used by KDE. nnn+nnn response means you want KDE to process each record using two printer lines.
nnnnn (create run only)	till [An] *	KSE prompts you for field descriptions until you have described the entire record.

Creating or Maintaining Format Descriptions of KDE Data Records (continued)

Keyword	Response	Considerations
nnnnn Hdd	nnn or nnn+nnn Command key 01 Command key 05 Command key 06 Command key 08	KSE prompts the next header line. You give lengths if you have more field descriptions to enter. KSE prompts correction sequence KSE prompts for deletion KSE prompts END-OF-JOB KSE prompts for included module
nnnnn	tlll [An] *	You keyed in record length. KSE now prompts you for field descriptions. When you complete the description statements for the entire record, KSE prompts you for another header line.
END-OF-JOB	YES or YES,START NO	If command keys 01, 05, or 08 were used, KSE displays and checks entire source library module for errors. If errors are found, KSE prompts the statement number so operator can correct the statement. If YES,START is entered, KSE prompts FORMAT DESCRIPTION for another job. KSE prompts the next statement number and header.

*Brackets ([]) denote an optional entry. They are not keyed as part of the response.

Creating or Maintaining Formatted KSE Source Statements (continued)

Keyword	Response	Considerations
END-OF-JOB	YES	KSE writes the source statements into the source library and ends the run.
	NO	KSE prompts for next statement number.
	YES,SERIAL or YES,RPG	KSE prompts SERIAL START POSITION.
	YES,START	After writing this module back in the source library, KSE prompts FORMAT DESCRIPTION for another job.
SERIAL START POSITION	Program start key	KSE inserts the statement number in positions 1-5 of each statement.*
	1-92	KSE inserts the statement number in each statement starting at the entered position.

*If YES,SERIAL was entered, all statements are serialized. This includes any tables following source statements.

If YES,RPG was entered, serialization stops when a statement with **Ø in positions 1-3 is read.

Creating or Maintaining Free-Form KSE Source Statements

Keyword	Response	Considerations
FORMAT DESCRIPTION	Program start key	You want to process free-form statements rather than a format description or source statements that require formats.
NEW SOURCE MODULE	YES	You are creating a new source library module.
	NO	You are maintaining an existing source library module.
SOURCE MODULE NAME	1-6 character name	You supply KSE with the name of the source library module you want to create or maintain.
SOURCE MODULE UNIT	F1, F2, R1, or R2	You supply KSE with the disk unit of the source library module you want to create or maintain.
WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.		
nnnnn NEW SOURCE STATEMENTS MAY BE ADDED TO THE SOURCE MODULE		
00000 (Creation) or nnnnn (Maintenance)	Free-form statement data	KSE displays the next incremented statement number. You key in source data.
	Command key 01	KSE prompts correction sequence.
	Command key 05	KSE prompts for deletion.
	Command key 06	KSE prompts END-OF-JOB.
	Command key 08	KSE prompts for included module.
END-OF-JOB	YES	KSE writes the source statements into the source library and ends the run.
	NO	KSE prompts for next statement number.
	YES,SERIAL or YES,RPG	KSE rpompts SERIAL START POSITION.
	YES,START	After writing this module back into the source library, KSE prompts FORMAT DESCRIPTION for another job.
SERIAL START POSITION	Program start key	KSE inserts the statement number in positions 1-5 of each statement.*
	1-92	KSE inserts the statement number in each statement starting at the entered position.*

*If YES,SERIAL was specified, all statements are serialized. This includes any tables following source statements. If YES,RPG was entered, serialization stops when a statement with **Ø in positions 1-3 is read.

Maintaining a Procedure

Keyword	Response	Considerations
FORMAT DESCRIPTION	Program start key	You want to process free-form statements rather than a format description or source statements that require formats.
NEW SOURCE MODULE	NO	You are maintaining an existing source library module.
SOURCE MODULE NAME	1-6 character name ,P	You supply KSE with the name of the source library module you want to maintain. <i>Note:</i> ,P indicates that you are going to maintain a procedure.
SOURCE MODULE UNIT	F1, F2, R1, or R2	You supply KSE with the disk unit of the source library module you want to maintain.
WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.		
nnnnn NEW SOURCE STATEMENTS MAY BE ADDED TO THE SOURCE MODULE		
nnnnn	Free-form statement data	KSE displays the next incremented statement number. You key in source data.
	Command key 01	KSE prompts correction sequence.
	Command key 05	KSE prompts for deletion.
	Command key 06	KSE prompts END-OF-JOB.
	Command key 08	KSE prompts for included module.
END-OF-JOB	YES	KSE writes the source statements into the source library and ends the run.
	NO	KSE prompts for next statement number.
	YES,START	KSE writes the source module into the source library and then prompts FORMAT DESCRIPTION to start another job.

INCLUDING KSE CONTROL STATEMENTS IN PROCEDURES

The KSE program is frequently used for entering the same type of data. Rather than go through the keyword-response sequence each time, the operator can include the information needed in control statements and include these control statements in an OCL procedure (OCL BUILD cycle). The procedure can then be called (OCL CALL cycle) by the operator, and KSE will get the necessary information from the control statements. Delayed responses (?) are allowed in the control statements. When you use chained procedures (OCL BUILD cycle) or press the enter minus key after responding to CALL NAME, the system will not prompt MODIFY after the OCL statements and again after the including statements, thus further reducing operator intervention. The BUILD, BUILD, and CALL cycles are described in the *IBM System/3 Model 6 Operation Control Language and Disk Utility Programs Reference Manual*, GC21-7516. Examples showing KSE control statements in procedures are included at the end of this chapter.

Entering Control Statements

Every control statement consists of an identifier and parameters. The identifier is the first word of the statement and for KSE is always //KSE in positions 1-7. Parameters are information you supply to the program. Every parameter consists of a keyword, which identifies the parameter, followed by the response, which you supply.

Coding Rules

The rules for writing control statements are as follows:

1. *Statement Identifier.* All control statements must have //KSE in positions 1-7.
2. *Statement Parameters.* Parameters can be in any order. Use a comma to separate one parameter from another. Use a hyphen (-) within each parameter to separate the keyword from the response. Do not use blanks within or between parameters.
3. *Statement Length.* Control statements must not exceed 96 characters. Each control statement may contain one or more parameters. (See following example.) Do not use a comma at the end of the control statement. A blank must follow the last response and indicates end of statement.

```

1   4   8   12  16  20  24  28  32
// KSE FDESC-YES, FTYPE-KSE, NEW-YES,
// KSE END

```

```

36  40  44  48  52  56  60  64  68
SNAME-?, SUNIT-?

```

OR

```

1   4   8   12  16  20  24  28  32
// KSE FDESC-YES, FTYPE-KSE
// KSE NEW-YES
// KSE SNAME-?, SUNIT-?
// KSE END

```

End Control Statement

The END statement is a special statement that indicates the end of control statements. It must contain //KSEEND in positions 1-10 and must always be the last control statement for the program.

Control Statement Diagnostics

The control statements are diagnosed by KSE as they are read from the procedure. Keywords having delayed responses (?) are prompted in conversational mode. If required parameters are not in the control statements, a message is printed and the keywords are prompted in conversational mode. If errors are detected in the control statements, an asterisk is printed under the error parameter, an error message is printed, and conversational mode is entered. The control statements are not updated by the information supplied in conversational mode.

Control Statement Summary Chart

The following chart shows the keywords and responses for the parameters in the KSE control statements. In addition to the indicated responses, the question mark (?) may be entered to specify a delayed response.

Keyword	Response	Considerations
FDESC	YES	You are processing format descriptions.
	NO	You are processing source statements rather than format descriptions.
	Program start key	You want to process free-form statements rather than a format description or source statements that require formats.
FTYPE	KDE	You are processing KDE format descriptions.
	KSE	You are processing KSE format descriptions.
FNAME	1-6 character name	You supply KSE with the name of the source library entry that contains the format descriptions.
	Program start key	You will not use format descriptions. All records will be free-form. This response is equivalent to pressing the program start key as the response to the FORMAT NAME prompt in conversational mode.
FUNIT	F1, F2, R1, or R2	You supply KSE with the disk unit number of the source library entry that contains the format descriptions.
DSPLY	YES or NO	You respond YES if you want KSE to display the format descriptions. You respond NO to bypass the display.
NEW	YES	You are creating a new source module
	NO	You want to maintain the statements in an existing source library module
SNAME	1-6 character name	You supply the name of the source library module you want to create or maintain.
	,P	<i>Note:</i> ,P indicates that you are going to maintain a procedure.
SUNIT	F1, F2, R1, or R2	You supply the unit of the disk on which the source library module you want to create or maintain is located.
FNUM	nn	You tell KSE which format description you want to use (nn is the 2-digit number in the header statement).

KSE ERROR MESSAGES

If an invalid response is keyed during the initialization prompt-response sequence or while entering or maintaining source modules, KSE will print an error message on the same line and reprompt the keyword. The following chart lists the KSE error messages and their meanings.

Message	Meaning
DELETED	All field descriptions after H99 are ignored.
DUPLICATE MODULE NAME (This error message is given only when creating a source library module.)	Response to SOURCE MODULE NAME has a match in the source library directory on the disk unit given as a response to SOURCE MODULE UNIT. KSE reprompts SOURCE MODULE NAME and SOURCE MODULE UNIT. These are the response options: <ol style="list-style-type: none"> 1. Rekey the same name with a different unit. 2. Rekey a different name with the same unit. 3. Rekey a different name with a different unit. 4. Rekey the same name and the same unit. This option causes KSE to replace the current source library module with data keyed in during this run. KSE prints a message indicating that the old module has been replaced.
ERRORS DETECTED – CONVERSATIONAL MODE ENTERED	While reading KSE control statements from a cataloged procedure, one or more of the following errors was detected. <ol style="list-style-type: none"> 1. <i>Duplicate Keywords.</i> A keyword appears more than once in the control statements. 2. <i>Punctuation.</i> The format of the control statements is wrong. An extra or missing comma is the most common type of error. A blank in or between parameters causes all characters following to be ignored (except following a comma). 3. <i>Keyword Spelling.</i> A keyword is spelled incorrectly. 4. <i>Response too Long.</i> The response to a keyword contains more characters than allowed. 5. <i>Invalid Identifier.</i> The control statement does not start with //bKSEb in positions 1-7. 6. <i>Invalid Response.</i> The response given is invalid for this keyword. An asterisk (*) is printed under the first position of the keyword or response in error.

KSE ERROR MESSAGES (continued)

Message	Meaning
FORMAT DESCRIPTION INCOMPLETE	<p>This message will be printed if:</p> <ol style="list-style-type: none"> 1. During a source run the format description used has insufficient fields to equal record length in the header statement. KSE will prompt for a different format. 2. During end-of-job processing for a format run a header is read before the previous format was fully described. KSE prompts the next statement number to allow the operator to enter field descriptions to complete the format.
FORMATS NOT FOUND	<p>Response to FORMAT NAME has no match in the source library directory for the disk unit you keyed as a response to FORMAT UNIT. KSE redisplay the format name and unit prompt.</p> <p>You have three options as responses to the format name and unit after the FORMATS NOT FOUND message:</p> <ol style="list-style-type: none"> 1. You can rekey the same name with a different disk unit. 2. You can rekey a different name with the same disk unit. 3. You can rekey a different name and disk unit.
HEADER NOT IN SEQUENCE	<p>The header records in a source module of Format Descriptions are not in sequence. This can occur if header statements are changed or deleted during maintenance.</p>
INVALID ACCUMULATOR NUMBER	<p>Only numbers 0 through 9 are valid accumulator numbers.</p>
INVALID ACCUMULATOR REFERENCE	<p>Response does not contain an A or blank in the sixth position.</p>
INVALID FIELD FOR ACCUMULATOR	<p>Only field types N, R, or P can reference accumulator.</p>
INVALID FIELD LENGTH	<p>Field length is too long for field type or exceeds forms width.</p>
INVALID FIELD TYPE	<p>Field type not N, R, A, D, or S for KSE or N, R, A, D, S, C, T, or P for KDE.</p>
INVALID FORMAT NAME	<p>Response to the prompt FORMAT NAME was invalid. KSE redisplay the prompt after printing the error message (see <i>Source Library Module Naming</i> in this chapter).</p>
INVALID FORMAT NUMBER	<p>Response to prompt SELECT FORMAT NUMBER was not valid two digit number, or header record requested was not found.</p>
INVALID FORMAT TYPE	<p>Response to FORMAT TYPE was not KSE or KDE. System re-prompts FORMAT TYPE.</p>

KSE ERROR MESSAGES (continued)

Message	Meaning
INVALID FORMAT UNIT	Response to the prompt FORMAT UNIT was invalid. F1, F2, R1, or R2 are the only valid responses. If you do not have a second disk drive on your system. F2 or R2 are invalid responses. KSE redisplay the prompt after printing the error message.
INVALID HEADER STATEMENT	Header statement contains non-digits in header number location.
INVALID MODULE	Response to the prompt SOURCE MODULE NAME was invalid. KSE redisplay this prompt after printing the error message (see <i>Source Library Module Naming</i> in this chapter).
INVALID MODULE UNIT	Response to the prompt SOURCE MODULE UNIT was invalid. KSE redisplay the prompt after printing the error message. F1, F2, R1 or R2 are the only valid responses.
INVALID RECORD LENGTH	<ol style="list-style-type: none"> 1. Record length of KDE record not valid format nnn or nnn+nnn. 2. Record length over 256 for KDE 3. Record length in header not 96 for KSE. 4. Record length not same as previous records. 5. Line length over 132.
INVALID SERIAL START POSITION	Response to SERIAL START POSITION was not numeric 1 through 92 or program start key.
INVALID STATEMENT NUMBER	Response to STATEMENT NUMBER was not five valid digits.
INVALID TOTAL LENGTH	Length given for the current field makes total length of all fields greater than record length or line length in header.
KEYWORD REQUIRED BUT NOT GIVEN – PROCESSING CONTINUED	A KSE control statement in a cataloged procedure does not contain a required keyword. The missing keyword is prompted in conversational mode.
KEYWORD WAS GIVEN BUT NOT USED – PROCESSING CONTINUED	<p>One or more of the following keywords were included in a cataloged procedure but were not needed: FUNIT, DSPLY, FNUM, FNAME, FTYPE.</p> <p>The keywords are ignored and processing continues. Example: FDESC-P/S, FUNIT-R1.</p>
LINE EXCEEDS PAGE WIDTH	Record length plus number of fields for a line is greater than width of forms.

KSE ERROR MESSAGES (continued)

Message	Meaning
<p>MAINTENANCE ONLY ALLOWED</p>	<p>This message will be printed if:</p> <ol style="list-style-type: none"> 1. While creating formats, format number H99 has been entered. 2. The source module being maintained already has 99 formats. <p>Command keys 01, 05, and 06 are allowed.</p>
<p>MISSING END STATEMENT</p>	<p>A cataloged procedure containing KSE control statements does not have an END control statement.</p>
<p>MODULE NOT FOUND</p> <p>(This error message is given only when maintaining an existing source library module or including a module using command key 08.)</p>	<p>Response to SOURCE MODULE NAME has no match in the source library directory on the disk unit you gave as the response to SOURCE MODULE UNIT. KSE redisplay the source name and unit prompt.</p> <p>You have four options for your responses:</p> <ol style="list-style-type: none"> 1. You can rekey the same name and a different disk unit. 2. You can rekey a different name and the same disk unit. 3. You can rekey a different name and a different disk unit. 4. You can rekey the same name and the same disk unit after changing packs on that unit.
<p>NO HEADER STATEMENT</p>	<p>When getting format descriptions for formatted source statements, a header record was expected but not found. System will prompt FORMAT NAME and FORMAT UNIT.</p>
<p>SELECT ONE COMMAND KEY ONLY</p>	<p>More than one command key pressed.</p>
<p>SOURCE LIBRARY FULL</p>	<p>No source statements may be entered because no source library exists on the referenced disk pack or the library is full.</p>
<p>WARNING: WORK SPACE HAS BEEN ALLOCATED DO NOT REMOVE ANY DISK PACKS.</p>	<p>This warning message is printed after the source module has been found. Because KSE uses available space on other disks as work areas, changing packs after the work areas are defined will cause data in the corresponding area of the new pack to be overlaid by the work area and lost.</p>
<p>WORK SPACE STILL IN USE. DO NOT REMOVE ANY DISK PACKS.</p>	<p>You have responded to END-OF-JOB with YES,START. The work space used for the first job will be used for any remaining jobs. Changing packs will cause data in the corresponding areas of the new pack to be overlaid and destroyed.</p>
<p>XXXXXX MODULE REPLACED</p>	<p>See option 4 under message DUPLICATE MODULE NAME.</p>

OCL LOAD CYCLE FOR KSE

Prompt	Response	Considerations
READY	LOAD	You wish to load a program (KSE) for execution.
LOAD NAME	\$KSE + (enter minus key)	Provide the name of the program to be executed.
UNIT	R1, R2, F1, or F2	Location of the disk in whose object library \$KSE resides.
MODIFY	RUN	Begin execution of the KSE program.

KSE SAMPLE JOBS

The format descriptions shown describe some of the more basic entries on RPG II File Description and Extension specifications.

KSE Sample Job 1. Create a Source Library Entry of KSE Format Descriptions

```

READY-                                LOAD
*****
010 LOAD          NAME-                $KSE
011              UNIT-                F1
*****
MODIFY

RUN

FORMAT DESCRIPTION ?                  YES
FORMAT TYPE -                        KSE
NEW SOURCE MODULE ?                  YES
SOURCE MODULE NAME -                 RPFMT
SOURCE MODULE UNIT -                 R1

**** WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.****

05736 NEW STATEMENTS MAY BE ADDED TO THE SOURCE MODULE

00000      H0196

00010      S05      (SKIP FIRST FIVE POSITIONS IN SOURCE STATEMENT)
00020      A09      (RPG SPECIFICATION TYPE, FILENAME)
00030      A01      (FILE TYPE)
00040      S03      (SKIP NEXT THREE POSITIONS)
00050      A01      (FILE FORMAT)
00060      R04      (BLOCK LENGTH)
00070      R04      (RECORD LENGTH)
00080      S12      (SKIP NEXT 12 POSITIONS)
00090      A07      (DEVICE)
00100      S50      (SKIP REMAINDER OF SPECIFICATION)

00110      H0296

00120      S05      (SKIP FIRST FIVE POSITIONS)
00130      A01      (RPG SPECIFICATION TYPE)
00140      S20      (SKIP NEXT 20 POSITIONS)
00150      A06      (TABLE NAME)
00160      S03      (SKIP NEXT THREE POSITIONS)
00170      R04      (NUMBER OF TABLE ENTRIES)
00180      R03      (LENGTH OF TABLE ENTRY)
00190      S54      (SKIP REMAINDER OF SPECIFICATION)
00200      H0396      (Press command key 06)

END OF JOB ?                          YES

KSE END OF JOB

```

KSE Sample Job 2. Create a Source Library Entry of Source Statements

These source statements will be RPG II File Description and Extension specifications and will use the format descriptions created in KSE Sample Job 1.

```

READY-                                LOAD
*****
010 LOAD          NAME-                $KSE
011              UNIT-                F1
*****
MODIFY

RUN

FORMAT DESCRIPTION ?                  NO
FORMAT NAME -                        RPGFMT
FORMAT UNIT -                         R1
DISPLAY FORMATS ?                    NO
NEW SOURCE MODULE ?                  YES
SOURCE MODULE NAME -                 PAYROL
SOURCE MODULE UNIT -                 R1

**** WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.****

00954 NEW STATEMENTS MAY BE ADDED TO THE SOURCE MODULE

SELECT FORMAT NUMBER -                01

      *  *      ** ** *  *          *  *
00000      FREPORT 0  F  1 132      TRACTR1
00010      FMASTER 0  F 512 128      DISK
00020
(Press command key 02)
SELECT FORMAT NUMBER -                02

      *  **          *  *  *  *  *
00020      E          ARR01      800 20
00030      E          ARR02      60 15
00040
(Press command key 06)
END OF JOB ?                          YES
KSE END OF JOB

```


KSE Sample Job No. 3. Maintain an Existing Source Library Entry Containing KSE Format Descriptions

The RPG II format descriptions from Sample Job 1 will be altered.

```

READY-                                LOAD
*****
010 LOAD          NAME-          SKSE
011              UNIT-          F1
*****
MODIFY

RUN

FORMAT DESCRIPTION ?                YES
FORMAT TYPE -                      KSE
NEW SOURCE MODULE ?                NO
SOURCE MODULE NAME -                RPGFMT
SOURCE MODULE UNIT -                R1

**** WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.****

05712 NEW STATEMENTS MAY BE ADDED TO THE SOURCE MODULE

00200      H0396                    (Press command key 01)
CORRECTING STATEMENT NUMBER -      130

00130      A01
00130      A09
CORRECTING STATEMENT NUMBER -      140

00140      S20
00140      S12
CORRECTING STATEMENT NUMBER -      (Press command key 06)
END OF JOB ?                        YES

```

```

00000      H0196
00010      S05          00130      A09
00020      A09          00140      S12
00030      A01          00150      A06
00040      S03          00160      S03
00050      A01          00170      R04
00060      R04          00180      R03
00070      R04          00190      S54
00080      S12          KSE END OF JOB
00090      A07
00100      S50
00110      H0296
00120      S05

```

KSE will provide a complete listing of format descriptions, with error checking, if command key 01, 05, or 08 is pressed during a maintenance run.

**KSE Sample Job 4. Maintain an Existing Source Library
Entry Containing Source Statements**

The RPG II source statements from Sample Job 2 will be altered.

```

READY-                                LOAD
*****
010 LOAD          NAME-                $KSE
011              UNIT-                F1
*****
MODIFY

RUN

FORMAT DESCRIPTION ?                  NO
FORMAT NAME -                        RPGFMT
FORMAT UNIT -                         R1
DISPLAY FORMATS ?                    NO
NEW SOURCE MODULE ?                  NO
SOURCE MODULE NAME -                 PAYROL
SOURCE MODULE UNIT -                 R1

**** WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.****

00952 NEW STATEMENTS MAY BE ADDED TO THE SOURCE MODULE
SELECT FORMAT NUMBER -                01

      * *      ** ** * *      * *
00040                                (Press command key 01)
CORRECTING STATEMENT NUMBER -        0
00000      FREPORT 0 F 1 132          TRACTR1
00000      FREPORT 0 F 150 210        TRACTR2
CORRECTING STATEMENT NUMBER -        (Press command key 02)
SELECT FORMAT NUMBER -                02

      * *      *      * * * * *
00040                                (Press command key 01)
CORRECTING STATEMENT NUMBER -        20
00020      E      ARR01      800 20
00020      EAFILE      ARR01      800 20
CORRECTING STATEMENT NUMBER -        (Press command key 06)
END OF JOB ?                          YES
KSE END OF JOB

```

**KSE Sample Job 5. Maintain a Source Library Entry
Containing KDE Format Descriptions**

Two format descriptions will be added to an existing entry.

```
READY-                                LOAD
*****
010 LOAD          NAME-                $KSE
011              UNIT-                F1
*****
MODIFY

RUN

FORMAT DESCRIPTION ?                 YES
FORMAT TYPE -                       KDE
NEW SOURCE MODULE ?                 NO
SOURCE MODULE NAME -                KDEFOR
SOURCE MODULE UNIT -                R1

**** WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.****

05700 NEW STATEMENTS MAY BE ADDED TO THE SOURCE MODULE

00030  H02                          100+100
00040  A100
00050  N005
00060  D050
00070  C015
00080  A030
00090  H03                          100+100
00100  N005
00110  A015
00120  N015
00130  T015
00140  R015
00150  A035
00160  A100
00170  H04                          (Press command key 06)

END OF JOB ?                         YES
KSE END OF JOB
```

**KSE Sample Job 6. Building a Procedure to Create a
New Source Module**

```
READY-                                BUILD
000 BUILD                             NAME-      PROCO1
001                                     UNIT-      R1
*****
010 LOAD                               NAME-      $KSE
011                                     UNIT-      F1
020 DATE                               -
030 SWITCH (00000000)                 -
040 FILE                               NAME-
*****
MODIFY

INCLUDE
*****
ENTER UTILITY CONTROL STATEMENTS
00

* CREATING A NEW SOURCE LIBRARY ENTRY      (This is a comment. Comments can be included in procedures
01                                         by entering an asterisk in position one.)

// KSE FDESC=NO,FNAME=RPGFMT,FUNIT=R1,DSPLY=NO,NEW=YES,SNAME=SLIB01,SUNIT=R1,FNUM=01
02

// KSE END                                ;
03

RUN
*****
MODIFY

RUN
```

KSE Sample Job 7. Calling Procedure Built by Sample Job 6

```
READY-          CALL  
000 CALL        NAME-      PROC01  
001             UNIT-      R1      (Press ENTER - to suppress printing of OCL procedure)
```

```
* CREATING A NEW SOURCE LIBRARY ENTRY  
// KSE FDESC=NO,FNAME=RPGFMT,FUNIT=R1,DSPLY=NO,NEW=YES,SNAME=SLIB01,SUNIT=R1,FNUM=01  
// KSE END
```

*** WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.***

01432 NEW STATEMENTS MAY BE ADDED TO THE SOURCE MODULE

```
      * *      ** ** * *      * *  
00000      FMASTER 0 F 512 128      DISK  
00010      FORMAT  0 F   1 132      TRACTR1  
00020      ACCOUNT 0 F 512 128      DISK  
00030      FILE    0 F  16 256      DISK  
00040      JOINT   0 F  16 256      DISK
```

00050 (Press command key 06)

END OF JOB ? YES

KSE END OF JOB

KSE Sample Job 8. Including Another Source Module and then Deleting Multiple Statements

```

READY-
000 CALL          NAME-          CALL
001              UNIT-          PROC03
001              UNIT-          R1
*****
010 LOAD          NAME-$KSE
011              UNIT-F1
*****
MODIFY

```

```

RUN
*****
INCLUDED STATEMENTS
00 // KSE FDESC-P/S,NEW-NO,SNAME-SLIB03,SUNIT-R1
01 // KSE END
*****
MODIFY

```

RUN

// KSE FDESC-P/S,NEW-NO,SNAME-SLIB03,SUNIT-R1

1 → // KSE END

**** WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.****

01420 NEW STATEMENTS MAY BE ADDED TO THE SOURCE MODULE

00060 (Press command key 08)

2 → SOURCE MODULE NAME - SLIB04

SOURCE MODULE UNIT - R1

INCLUDE AFTER STATEMENT NUMBER - 20

00030 FREE FORMAT STATEMENT NUMBER 6

00040 FREE FORMAT STATEMENT NUMBER 7

00050 FREE FORMAT STATEMENT NUMBER 8

00060 FREE FORMAT STATEMENT NUMBER 9

3 → 00070 FREE FORMAT STATEMENT NUMBER 10

00110 (Press command key 05)

DELETING STATEMENT NUMBER - (Press program start key)

4 → FROM - 40 TO - 60

DELETING STATEMENT NUMBER - (Press command key 06)

END OF JOB ? YES

KSE END OF JOB

1

Contents of SLIB03 at start of Sample Job 8.

FREE FORM STATEMENT NUMBER 0
FREE FORM STATEMENT NUMBER 1
FREE FORM STATEMENT NUMBER 2
FREE FORM STATEMENT NUMBER 3
FREE FORM STATEMENT NUMBER 4
FREE FORM STATEMENT NUMBER 5

2

Contents of SLIB04 at start of Sample Job 8

FREE FORM STATEMENT NUMBER 6
FREE FORM STATEMENT NUMBER 7
FREE FORM STATEMENT NUMBER 8
FREE FORM STATEMENT NUMBER 9
FREE FORM STATEMENT NUMBER 10

3

SLIB03 in work area (with statement numbers in parentheses) after command key 8 is used to include SLIB04. The statements after the point of inclusion are resequenced.

(00000) FREE FORM STATEMENT NUMBER 0
(00010) FREE FORM STATEMENT NUMBER 1
(00020) FREE FORM STATEMENT NUMBER 2
(00030) FREE FORM STATEMENT NUMBER 6
(00040) FREE FORM STATEMENT NUMBER 7
(00050) FREE FORM STATEMENT NUMBER 8
(00060) FREE FORM STATEMENT NUMBER 9
(00070) FREE FORM STATEMENT NUMBER 10
(00080) FREE FORM STATEMENT NUMBER 3
(00090) FREE FORM STATEMENT NUMBER 4
(00100) FREE FORM STATEMENT NUMBER 5

4

SLIB03 in work area (with statement numbers in parentheses) after command key 5 is used to delete statements.

(00000) FREE FORM STATEMENT NUMBER 0
(00010) FREE FORM STATEMENT NUMBER 1
(00020) FREE FORM STATEMENT NUMBER 2
(00030) FREE FORM STATEMENT NUMBER 6
(00070) FREE FORM STATEMENT NUMBER 10
(00080) FREE FORM STATEMENT NUMBER 3
(00090) FREE FORM STATEMENT NUMBER 4
(00100) FREE FORM STATEMENT NUMBER 5

**KSE Sample Job 9. Deleting Multiple Source Statements
and then Including Another Source Module**

```
READY-          CALL
000 CALL        NAME-      PROC03
001             UNIT-      R1 (Press ENTER - key)

// KSE FDESC-P/S,NEW-NO,SNAME-SLIB03,SUNIT-R1
// KSE END

**** WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.****

01420 NEW STATEMENTS MAY BE ADDED TO THE SOURCE MODULE

1 → 00060 (Press command key 05)
    DELETING STATEMENT NUMBER - (Press Program Start key)
2 → FROM -          30    TO -          40
    DELETING STATEMENT NUMBER - (Press command key 08)
3 → SOURCE MODULE NAME -          SLIB04
    SOURCE MODULE UNIT -          R1
    INCLUDE AFTER STATEMENT NUMBER - 10
00020    FREE FORMAT STATEMENT NUMBER 6
00030    FREE FORMAT STATEMENT NUMBER 7
00040    FREE FORMAT STATEMENT NUMBER 8
00050    FREE FORMAT STATEMENT NUMBER 9
00060    FREE FORMAT STATEMENT NUMBER 10
00090    (Press command key 06)
4 → END OF JOB ?
    KSE END OF JOB
```


1

Contents of SLIB03 at start of Sample Job 9

FREE FORM STATEMENT NUMBER 0
FREE FORM STATEMENT NUMBER 1
FREE FORM STATEMENT NUMBER 2
FREE FORM STATEMENT NUMBER 3
FREE FORM STATEMENT NUMBER 4
FREE FORM STATEMENT NUMBER 5

2

SLIB03 in work area (with statement numbers in parentheses)
after command key 5 is used to delete statements. The statements
are not resequenced after deletion.

(00000) FREE FORM STATEMENT NUMBER 0
(00010) FREE FORM STATEMENT NUMBER 1
(00020) FREE FORM STATEMENT NUMBER 2
(00050) FREE FORM STATEMENT NUMBER 5

3

Contents of SLIB04 at start of Sample Job 9

FREE FORM STATEMENT NUMBER 6
FREE FORM STATEMENT NUMBER 7
FREE FORM STATEMENT NUMBER 8
FREE FORM STATEMENT NUMBER 9
FREE FORM STATEMENT NUMBER 10

4

SLIB03 in work area (with statement numbers in parentheses)
after command key 8 is used to include SLIB04. The statements
after the point of inclusion are resequenced.

(00000) FREE FORM STATEMENT NUMBER 0
(00010) FREE FORM STATEMENT NUMBER 1
(00020) FREE FORM STATEMENT NUMBER 6
(00030) FREE FORM STATEMENT NUMBER 7
(00040) FREE FORM STATEMENT NUMBER 8
(00050) FREE FORM STATEMENT NUMBER 9
(00060) FREE FORM STATEMENT NUMBER 10
(00070) FREE FORM STATEMENT NUMBER 2
(00080) FREE FORM STATEMENT NUMBER 5

KSE Sample Job 10. Correcting Statements in Two Source Modules in one KSE Run

```

READY-                                LOAD
*****
010 LOAD          NAME-                SKSE
011              UNIT-                F1
*****
MODIFY

RUN

FORMAT DESCRIPTION ?                  NO
FORMAT NAME -                          RPGFMT
FORMAT UNIT -                          R1
DISPLAY FORMATS ?                      NO
NEW SOURCE MODULE ?                    NO
SOURCE MODULE NAME -                   SLIB01
SOURCE MODULE UNIT -                   R1

**** WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.****

01426 NEW STATEMENTS MAY BE ADDED TO THE SOURCE MODULE
SELECT FORMAT NUMBER -                 01

* * * * *
00050 (Press command key 01)
CORRECTING STATEMENT NUMBER -          10
00010      FORMAT  0  F  1 132          TRACTR1
00010      FORMAT  0  F  1 132          TRACTR2
CORRECTING STATEMENT NUMBER - (Press command key 06)
END OF JOB ?                          YES,START Tells KSE that another KSE job is to be run.
KSE END OF JOB
**** WORK SPACE STILL IN USE. DO NOT REMOVE ANY DISK PACKS. ****

FORMAT DESCRIPTION ?                  NO
FORMAT NAME -                          RPGFMT
FORMAT UNIT -                          R1
DISPLAY FORMATS ?                      NO
NEW SOURCE MODULE ?                    NO
SOURCE MODULE NAME -                   SLIB02
SOURCE MODULE UNIT -                   R1

01426 NEW STATEMENTS MAY BE ADDED TO THE SOURCE MODULE
SELECT FORMAT NUMBER -                 02

* * * * *
00050 (Press command key 01)
CORRECTING STATEMENT NUMBER -          10
00010      E                ARR04        2 10
00010      E                ARR04       120 10
CORRECTING STATEMENT NUMBER - (Press command key 06)
END OF JOB ?                          YES
KSE END OF JOB

```

**KSE Sample Job 11. Maintaining One Source Module and
Creating Another in One KSE Run Using Cataloged KSE
Control Statements**

```
READY-
000 CALL          NAME-      CALL
001              UNIT-      PROC02
                      R1 (Press ENTER- key to suppress printing of OCL procedure)
```

* MAINTAINING AN EXISTING SOURCE LIBRARY ENTRY AND CREATING A NEW SOURCE LIBRARY ENTRY

```
// KSE FDESC=NO,FNAME=RPGFMT,FUNIT=R1,DSPLY=NO,NEW=?,SNAME=?,SUNIT=R1,FNUM=01
// KSE END
```

} Cataloged
control
statements

```
NEW SOURCE MODULE ?          NO } Operator must respond to prompts
SOURCE MODULE NAME -        SLIB01 } with delayed responses
```

**** WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.****

01428 NEW STATEMENTS MAY BE ADDED TO THE SOURCE MODULE

* * * * *

00050 (Press command key 01)

```
CORRECTING STATEMENT NUMBER -      40
00040      JOINT      0  F  16 256      DISK
00040      JOINT      0  F  512 128      DISK
```

CORRECTING STATEMENT NUMBER - (Press command key 06)

```
END OF JOB ?          YES,START
KSE END OF JOB
```

**** WORK SPACE STILL IN USE. DO NOT REMOVE ANY DISK PACKS. ****

```
NEW SOURCE MODULE ?          YES } The control statements used in the first job are used again with
SOURCE MODULE NAME -        SLIB02 } different operator responses to prompts with delayed responses.
```

01428 NEW STATEMENTS MAY BE ADDED TO THE SOURCE MODULE

* * * * *

00000

SELECT FORMAT NUMBER - 02

* * * * *

```
00000      E          ARR03      20  5
00010      E          ARR04      2 10
00020      E          ARR05     200 20
00030      E          ARR06     35 60
```

```
00040      E          ARR07     440 25
```

00050 (Press command key 06)

```
END OF JOB ?          YES
KSE END OF JOB
```

**KSE Sample Job 12. Calling a Procedure Containing Errors
in KSE Control Statements**

```

READY-
000 CALL          NAME-          CALL
001              UNIT-          PROC04
                                R1 (Press ENTER- to suppress printing of OCL procedure)

// KSE FDES1D-NO,FNAME-RPGFRMT,FUNIT-T21
* 1 * 2 * 3

//KSE D4SPLY-NO,NEW-YES5SNAME-SLIB05,SUNIT7R1, FNUM-01
* 4 * 5 * 6 * 7 * 8

// KSE END

*** ERRORS DETECTED--CONVERSATIONAL MODE ENTERED ***

FORMAT DESCRIPTION ?          NO
FORMAT NAME -                 RPGFMT
FORMAT UNIT -                 R1
DISPLAY FORMATS ?            NO
NEW SOURCE MODULE ?          YES
SOURCE MODULE NAME -         SLIB05
SOURCE MODULE UNIT -         R1

**** WARNING: WORK SPACE HAS BEEN ALLOCATED. DO NOT REMOVE ANY DISK PACKS.****

01418 NEW STATEMENTS MAY BE ADDED TO THE SOURCE MODULE
SELECT FORMAT NUMBER -       01
* * * * * * * * * * * * * * * *
00000      FREPORT  O  F  1 132      TRACTR1
00010      FREPORT  O  F 150 210      TRACTR2
00020 (Press command key 06)
END OF JOB ?                 YES
KSE END OF JOB

```

- 1 Keyword spelling. FDES¹D should be FDESC.
- 2 Response too long. Only six characters are allowed in the format name.
- 3 Invalid response. T1 is not a valid unit.
- 4 Invalid identifier. The blank between // and KSE is missing.
- 5 Position 8 is flagged because a keyword does not start here. The first keyword must start in position 8. Error 4 caused this error.
- 6 Invalid response. Because the comma is missing between parameters, YES⁵SNAME is flagged as an invalid response to the keyword NEW.
- 7 Punctuation error. The comma between keyword and response is missing.
- 8 Invalid response. A keyword cannot start with a blank.

The Keyboard Data Entry utility program is used to create and maintain data files. The operator keys in the data fields on the keyboard. KDE uses format description statements from a source library module created by KSE to control the processing of the input data. The KDE program writes the input data into the data file on disk.

Note: KDE uses KSE modules, therefore, KSE and KDE must be on the same pack.

KDE DATA FILES

KDE data files can contain any type of data. You can use KDE to create or maintain KDE files which contain data for billing, payroll, sales, accounts payable, or any other processing application. The KDE data files are used as input and update files by your processing programs.

Data files are made up of a variable number of data records. The length of a data record can be up to 256 characters.

A KDE data file may contain more than one type of record. You may have groups of data records with various field sequences in one KDE data file. However, all of the records in a file must be the same length even when the format of the records is different.

All data files processed by KDE must have the file name of KDEFILE. Many different KDE data files can be used by varying the file label and date.

KDE does not prompt for information about the KDE-FILE. KDEFILE must be defined in the file keyword prompting portion of the OCL cycle that loaded KDE.

KDE does not support multivolume files even though the HIKEY parameter is allowed in the OCL sequence. At the end of the first volume, KDE issues the message YOU HAVE FILLED FILE. You can maintain each volume independently as long as you know the HIKEY parameters for each volume.

KDE files are processed as indexed files. Unlike KSE, which assigns statement numbers but does not put them in the source library as part of the source statement, KDE uses a key field which is part of the data record to index records in the file.

Key Fields

A key field must be defined in each KDE record. The key field can be from 1 to 29 characters long and may be made up of more than one data field.

All records in a data file must have the key field in the same location, even when the format of the records varies.

The key field may overlap other fields in a data record. However, a packed field (P-type) cannot be part of the key field unless the entire key field is described by one packed field or the least significant (rightmost) portion of one packed field. If a key field includes the units position of a N (numeric) field, negative data cannot be entered in that field.

R-type key fields are valid when the operator supplies the keys. However, the entire field must be entered using valid numeric digits (not blanks).

The operator defines the key field by responding to the prompts KEY FIELD START and KEY FIELD LENGTH. The responses are checked for validity (see *KDE Error Messages; Invalid Key Field* in this chapter).

If the operator responds to KEY FIELD START with NO (or NO, P for packed key field), KDE will assign the first five (three if packed key) bytes of each record as the key field. As each file is created, KDE fills the key field with a 5-digit number starting with 00000 and increments it by 10 for each record.

To have KDE assign and fill the key field, the format descriptions used for that file must have N005 or A005 (P003 for packed key field) as the first field description. An R-type field is not valid.

KDE RECORD FORMAT DESCRIPTIONS

The KDE utility processes KDE files using format control. Format descriptions of KDE data records are created and maintained using KSE (see *Chapter 2. Format Descriptions of KDE Data Records*).

KDE INITIALIZATION KEYWORD-RESPONSE SEQUENCE

During the initialization keyword-response sequence the operator provides KDE with information needed to process KDEFIL (see *KDE Keyword-Response Summary Chart* in this chapter).

By responding to **FORMAT NAME** and **FORMAT UNIT** the operator tells KDE which source library module contains the format descriptions to be used during this run. If the response to **DISPLAY FORMATS** is **YES**, KDE will list all format descriptions in that module. All formats in the module will be checked for validity. If an error is found, a message will be printed next to the format in error (see

Chapter 2. KSE Error Messages) and KDE will reprompt **FORMAT NAME**. The operator can either select a different format description module or press command key **06** and end the job. Errors in format descriptions must be corrected using KSE before any format descriptions in that source library module can be used by KDE.

KDE then prompts **NEW KDE FILE** to determine if this is a creation (**YES** response) or maintenance (**NO** response) run.

After prompting **KEY FIELD START** and **KEY FIELD LENGTH** (see *Key Fields* in this chapter), KDE prompts **SELECT FORMAT NUMBER** for the number of the format description to be used.

KDE Keyword-Response Summary Chart

The following chart shows the keyword-response sequence for creating or maintaining a KDE data file:

Keyword	Response	Considerations
FORMAT NAME	1-6 character name	You supply KDE with the name of the source library entry that contains the KDE format descriptions.
	Program start key	You are entering free-form records. The next prompt is NEW KDE FILE.
FORMAT UNIT	F1, F2, R1, or R2	You supply KDE with the disk unit number of the source library entry that contains the KDE format descriptions.
DISPLAY FORMATS	YES or NO	You respond YES if you want KDE to display the format descriptions. You respond NO to bypass the display.
NEW KDE FILE	YES	You are creating a new KDE file. Next prompt is KEY FIELD START.
	NO	You are going to maintain an existing KDE file.
KEY FIELD START	nnn	You furnish KDE with the starting location of the key field.
	NO	KDE will use the first five bytes of each record as the key field, and enter a number starting with 00000 and incremented by 10 for each record. KEY FIELD LENGTH will not be prompted.
	NO,P	KDE will use the first three bytes of each record as the key field, and enter a packed decimal number starting with 00000 and incremented by 10 for each record.
KEY FIELD LENGTH	nn	Length of key field.
SELECT FORMAT NUMBER	nn	You tell KDE which format description you want to use (nn is the 2-digit number in the header statement).

KSE displays the field start locations. KSE prints one line with an asterisk (*) in each field starting print position.

* * * * *

You then key in the data fields for the record, or press a command key for the processing function you want.

ENTERING DATA RECORDS

After the operator has selected a format description, KDE prints an asterisk in the starting position of each field. The operator then keys the data into the record one field at a time, pressing the program start key (or enter minus key for negative fields) after each field.

Alphameric Fields

Alphameric (A-type) fields are printed as the characters are entered. If the operator presses the program start key (indicating end of field) before entering the entire field, the remaining positions are filled with blanks. Alphameric fields (A, C, T, and D types) can be duplicated one character at a time by using the tab key. As long as the tab key is held depressed, characters in the field are duplicated from the previous record. At the end of the field, the program waits until the tab key is released. The program start key must then be pressed to enter the field into the record.

Signed Numeric Fields

Signed numeric (N-type) fields are not printed as they are entered. Because signed numeric fields are right-justified and the function key (enter plus or enter minus) determines the sign, the field is printed after the end of the field is indicated. The sign is printed following the units position of the field. To zero-fill a signed numeric field, the operator must enter at least one zero. If no data is entered (only the program start key is pressed), the field is blank-filled except for a zero in the low order position.

Unsigned Numeric Fields

Unsigned numeric (R-type) fields are not printed as they are entered. Because they are right justified, the characters are not printed until the end of the field is indicated. The leading positions are blank-filled. The program start key or the enter plus key can be used to enter the data.

Automatic Duplicate Fields

D-type fields are automatically duplicated from the previous record if the automatic duplication indicator is on. Field light 03 indicates the condition of the automatic duplication indicator.

When a different format is selected, automatic duplication is suppressed for the first record of the new format. However, field light 03 still shows the condition of the automatic duplication indicator.

Skip Fields

S-type fields are filled with blanks automatically.

Packed Fields

P fields are packed decimal fields. Numeric data entered is packed two characters to each byte in the data record.

Self-Check Fields

The KDE utility supports modulus 10 and modulus 11 self-check fields. The T field type in the field description statement tells KDE that this field should be self-checked using modulus 10. The C field type in the field description statement tells KDE that this field should be self-checked using modulus 11. A self-check field cannot be greater than 15 characters. The self-check digit (right-most character keyed) must be numeric. Self-check fields are processed as alphameric and therefore must be right-justified by the operator.

KDE processes the self-check data and compares the result with the self-check digit. KDE displays an error message if the result is not equal to the self-check digit. The print head is then repositioned by KDE to allow the operator to re-key the self-check data.

Modulus 11

To determine manually the check digit for modulus 11 self-check data:

1. Apply the weights 2, 3, 4, 5, 6, and 7 successively from the low-order digit to the high-order digit of the self-check data. If the data contains more than six digits, repeat the weight assignment, starting with the weight of 2 on the seventh position.
2. Multiply each digit of the data by its assigned weight.
3. Add the developed products together to arrive at the sum of the products.
4. Divide the sum of the products by 11.
5.
 - a. When the sum of the products is evenly divisible by 11, the check digit is 0.
 - b. When the sub of the products is not evenly divisible by 11, subtract the remainder from 11. The difference is the check digit.

The following chart illustrates the manual method of determining the modulus 11 check digit.

Self-check data	4	5	6	2	6	7
Multiply each position by the proper weight	7	6	5	4	3	2
Add the products	28 + 30 + 30 + 8 + 18 + 14 = 128					
Divide the sum of the products by 11	128 ÷ 11 = 11 with a 7 remainder					
Subtract the remainder from 11						11
						-7
						<u>4</u>
The difference is the check digit						
Self-checking number	4562674					

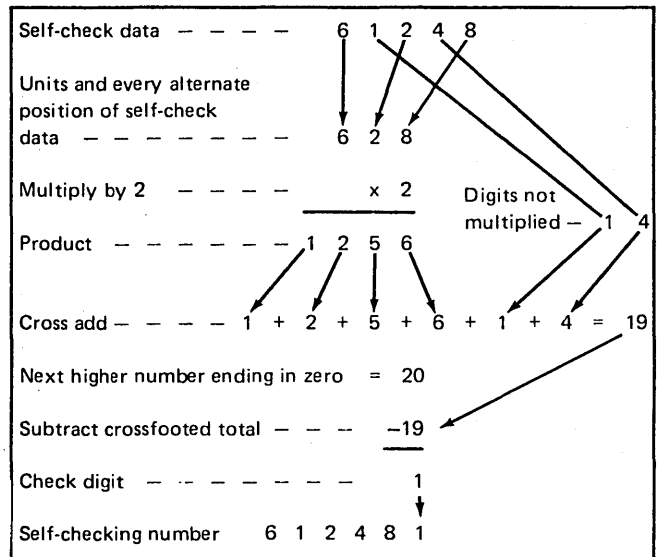
Note: The object of the calculation is to arrive at a number that, when added to the sum of the products, is evenly divisible by 11. When using this system, every eleventh number cannot be used because dividing the sum of the products by 11 results in a remainder of 1 and this requires a check digit of 10. Because a check digit of 10 occupies two positions on the document, it cannot be used.

Modulus 10

To determine the check digit for modulus 10 self-check data:

1. Multiply the units position and every alternate position of the self-check data by 2.
2. Cross add the digits in the product and the digits in the self-check data that are not multiplied by 2.
3. Subtract the crossfooted total from the next higher number ending in zero.
4. The difference is the check digit.

The following chart illustrates the method of determining the modulus 10 check digit.



Free-Form Records

Free-form records are data records for which no format descriptions have been supplied. The operator can enter free-form records by pressing the program start key in response to FORMAT NAME. Because there is no format description header statement to determine record length, the length is set at 96 characters.

The 96-character record is divided into fields depending on the operator's response to KEY FIELD START.

Response to KEY FIELD START	Type of Fields in Record
Valid Number 1-96	One 96-character A-type field in each record with the key field included.
NO	One 5-byte N-type field (KDE supplied key field) and one 91-character A-type field.
NO, P	One 3-byte P-type field (KDE supplied 5-character key field packed into 3 bytes) and one 93-character A-type field.

FORMAT UNIT, DISPLAY FORMATS, and SELECT FORMAT NUMBER will not be prompted.

Free-form records can also be entered when entering formatted data records by responding to SELECT FORMAT NUMBER with 00. The same rules on field breakdown apply, except record length is the same as the record length from the format descriptions. Free-form will only be allowed if record length plus number of fields (plus 2 if KDE supplies packed key field) does not exceed page width.

KDE ACCUMULATORS

Accumulators provide you with a means of verifying the accuracy of the data keyed into the KDE file before you use that file in a processing program. The accumulator displays can be compared with control totals for verification.

Batch Total Accumulators

The KDE utility program provides ten accumulators with which you can maintain batch totals during a processing run. These accumulators are numbered 0-9 and are referenced in the field description statement for a field. The accumulator response [An] after the field length in the field description tells KDE to add the contents of that field to the specified accumulator. Only numeric (N or R type) or packed (P type) fields can be accumulated. More than one field from a data record can be added to the same accumulator. The fields need not be the same type or length. When command key 08 is pressed, KDE displays the contents of the batch accumulators. To prevent all zero totals from printing, command key 08 cannot be used until an accumulator is referenced. The batch accumulators are also displayed (if referenced) when command key 06 is pressed. The accumulators are set to zero after being displayed.

Final Total Accumulators

KDE also provides a corresponding group of ten final total accumulators. These accumulators are not directly updated by the data keyed into fields. When command key 08 is used to total the batch accumulators, the values from the batch accumulators are added to the final accumulators. Each time the batch accumulators are displayed, the final accumulators are updated. When command key 06 is pressed, the batch accumulators are displayed and added to the final accumulators before the final accumulators are displayed. If no accumulators were referenced during the KDE job, no accumulators are displayed.

Updating Accumulators

The batch accumulators are updated whenever they are referenced by a field description. If fields which reference accumulators are duplicated by using command key 04 or 07, the accumulators are still updated. Changing format descriptions does not change the accumulator totals.

When command key 01 is used to correct a field in a previous record, the replaced data is subtracted from the accumulators, and the new data is then added. For example, if accumulator 01 contains a value of 750 before the record is corrected, a field in the correction cycle that references accumulator 01 is changed from 50 to 100. At the end of the correction cycle, accumulator 01 contains 800. Likewise, if the accumulator started at -600 and the value of the field changed from 100 to -150, the accumulator would contain -850.

Note: If formats are changed during the correction cycle, the accumulators referenced by the new field descriptions would be updated when the record is corrected.

When KDE records are deleted using command key 05, the current format descriptions are used to update accumulators. For any field descriptions that reference accumulators, the data in the corresponding positions of the record being deleted is subtracted from the accumulator totals.

KDE COMMAND KEYS

KDE command keys perform the following functions:

<i>Command Key</i>	<i>Function</i>
01	Correct a previous record.
02	Select a different format description.
03	Change the contents of an automatic duplication field. The condition of the automatic duplication indicator and of field light 03 is reversed.
04	Duplicate a field
05	Delete a record.
06	Print batch and final accumulator totals and end the job.
07	Duplicate the remainder of this record.
08	Print batch accumulator totals.
03 & 04	Re-enter the current record.
03 & 07	Correct a previous field in the current record.
04 & 07	Skip the remainder of this record.

The field lights are used in conjunction with command key operations. Field light 03 indicates the condition of the automatic duplication indicator. If field light 03 is on, the indicator is on. The other field lights indicate when command keys can be used. When a particular field light is on, its corresponding command key can be used.

Correcting Current Record

If the operator makes a mistake keying in a field and realizes it before indicating end of field (by pressing program start, enter plus, or enter minus), the BKSP function key will backspace one character for each depression allowing the operator to re-enter those characters. The FIELD ERASE function key will return to the start of the current field. The operator can correct or change previously entered fields in the current record by pressing command keys 04 and 03 or 07 and 03. Both combinations return the carriage to the first field, but the 07, 03 combination also moves the current record (as much as has been entered) to the previous record area. The operator then can correct fields or use command key 04 to duplicate fields that are correct. The setting of the automatic duplication indicator is not changed.

The command key combinations are not allowed before the first field in a record. If pressed at this time KDE will print TOO MANY COMMAND KEYS and allow the first field to be entered again.

Correcting Previous Record

To change a previous record the operator presses command key 01. KDE then prompts CORRECTING RECORD KEY. If KDE finds the record with the key field supplied by the operator, it prints the record using the current format description, moves it to the previous record area, and allows the operator to key in the new data. When this record is completely processed, KDE prompts CORRECTING RECORD KEY for the next record.

When a negative numeric field ending in zero is displayed in alphameric mode (the current format description specifies alphameric mode), the field should be re-entered only by duplication or the sign of the field may change. Example: A numeric field containing -120 (hexadecimal F1F2D0) is displayed in alphameric mode as 12&. If 12& is keyed in, hexadecimal F1F250 is entered, and the sign of the field may change.

While in a correcting or deleting cycle, command key 02 can be used to select a different format. Command key 08 and 06 are also allowed (see *KDE Command Key Usage*).

Pressing command key 01 will cause KDE to return to normal processing.

Whenever KDE prompts for record key (either **CORRECTING RECORD KEY**, or **DELETING RECORD KEY**), the response is processed as an alphameric or numeric field. Thus, the entire record key must be keyed in exactly as it was originally entered, including leading zeros or blanks.

Deleting Previous Record

A previous record can be deleted during a creation or maintenance run. To delete a record the operator presses command key 05. KDE prompts **DELETING RECORD KEY**. The operator responds with the key field of the record that is to be deleted. If KDE finds that record it fills the entire record, except for the key field, with blanks. KDE then prompts **DELETING RECORD KEY** for the next record to be deleted. Accumulators are updated using the current format descriptions.

Skipping Rest of Record

The operator can skip the remaining fields of a record by pressing command keys 04 and 07. The remaining fields will be filled as follows:

A (Alphameric) – blanks

N (Signed numeric) – zeros

R (Unsigned numeric) – blanks

P (Packed) – zeros and then packed

C & T (Self-check) – blanks (self-checking is bypassed)

D (Automatic Duplicate) – blanks if automatic duplication indicator (field light 03) is off.

D (Automatic Duplicate) – duplicated from previous record if automatic duplication indicator (field light 03) is on.

S (Skip) – blanks

KDE Command Key Usage

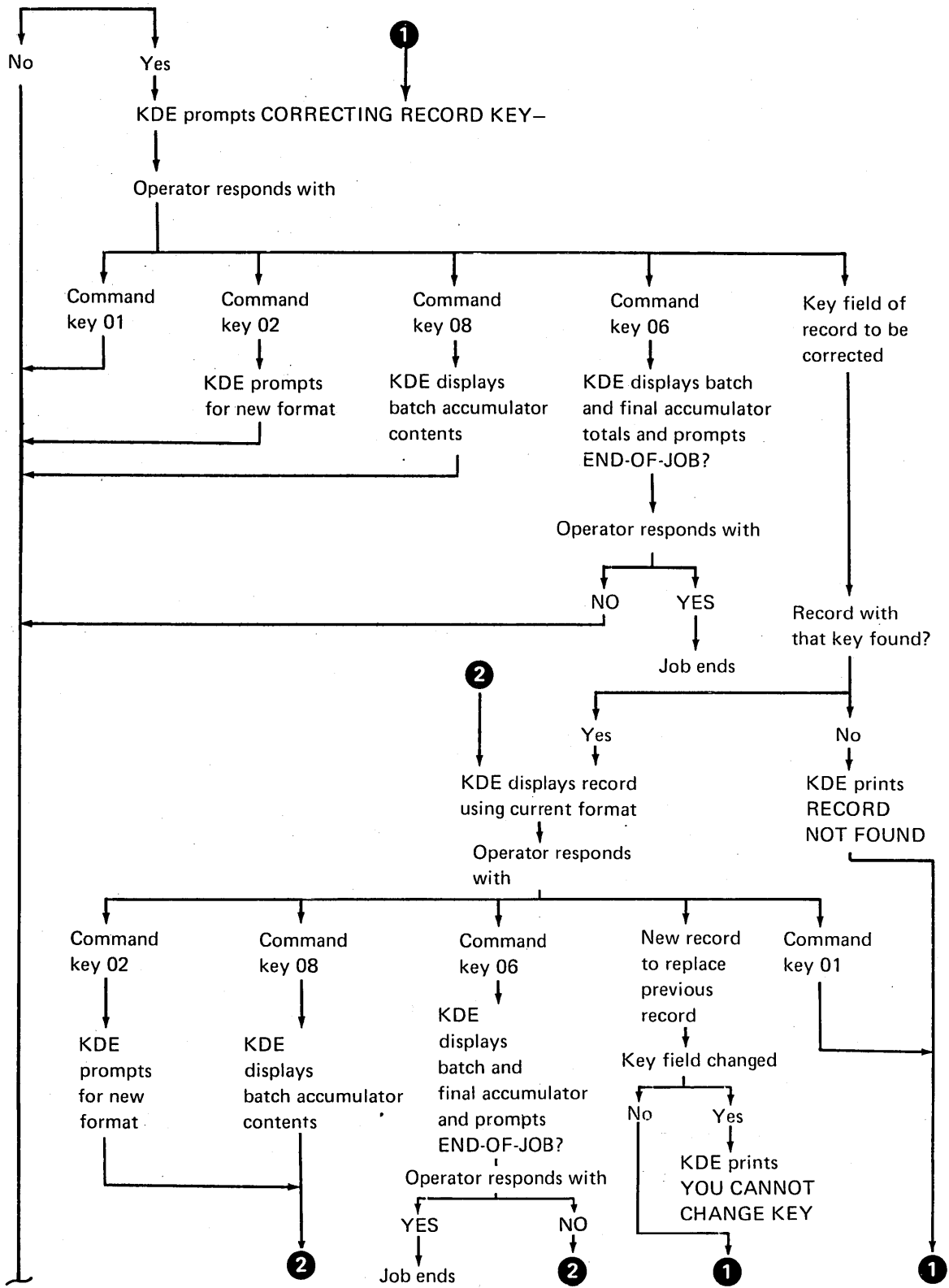
The following chart shows the operator-system interaction when the operator is using command keys.

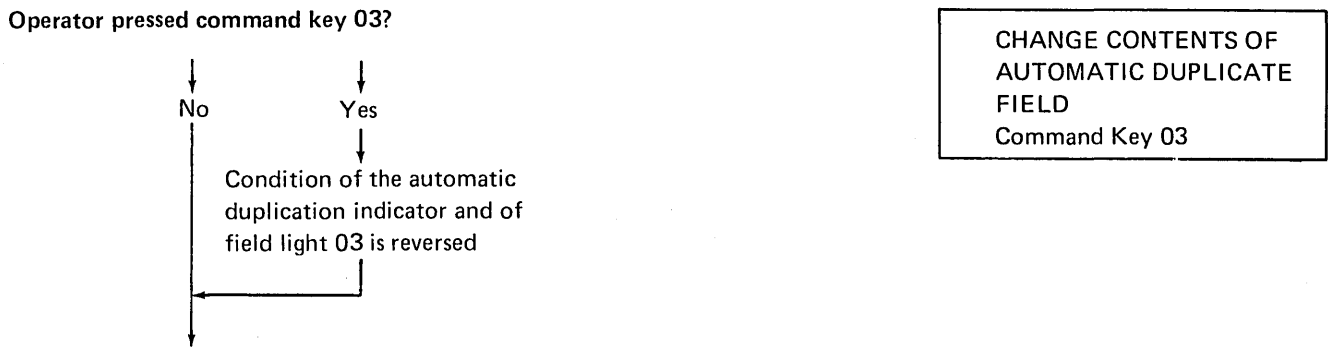
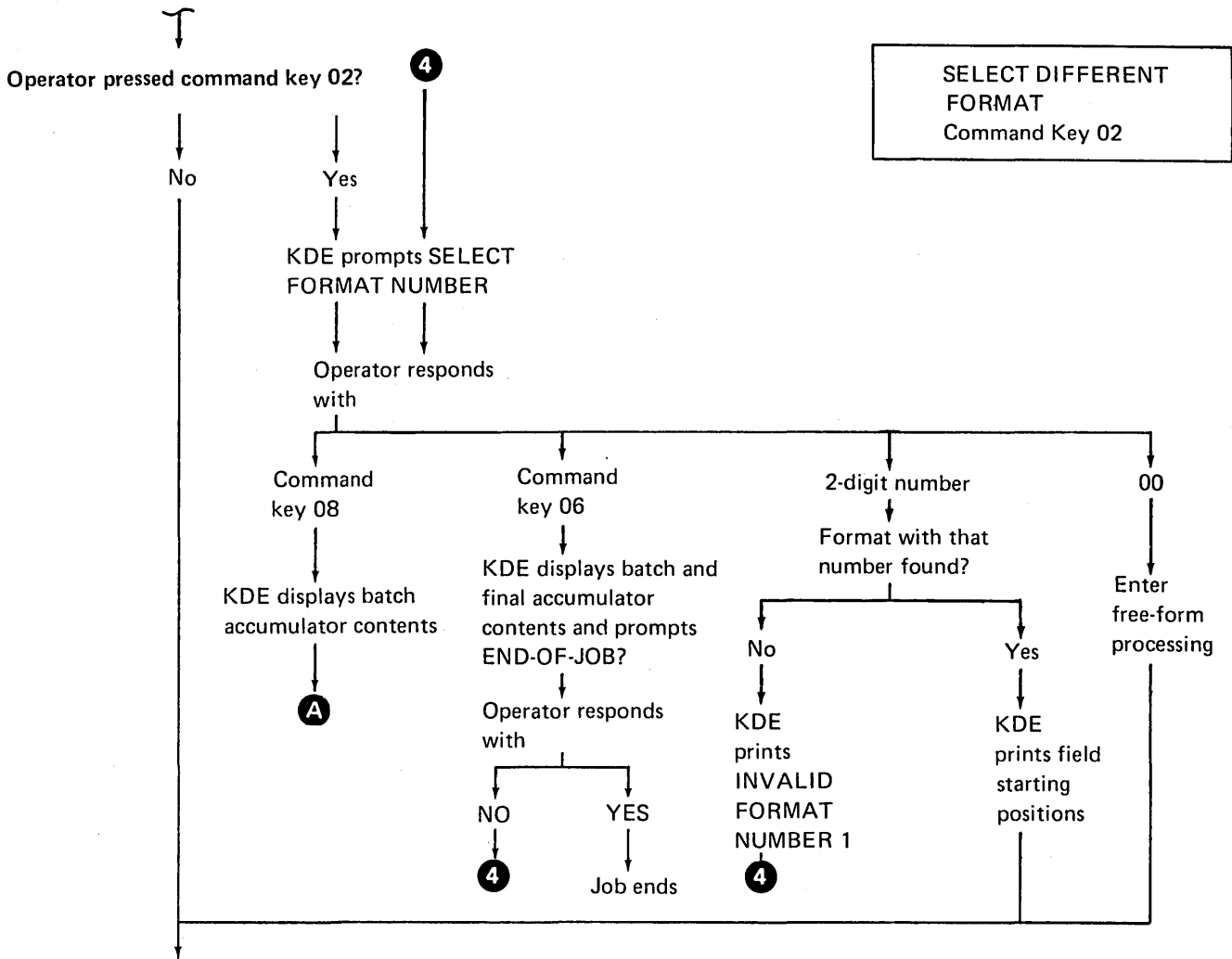
Normal Addition Processing

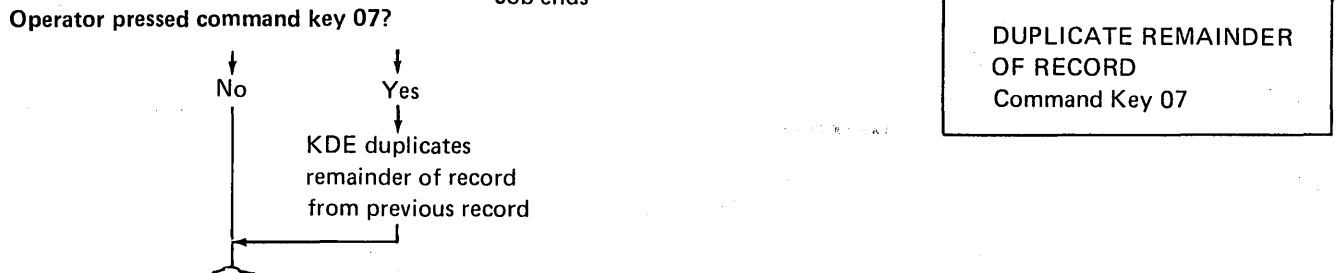
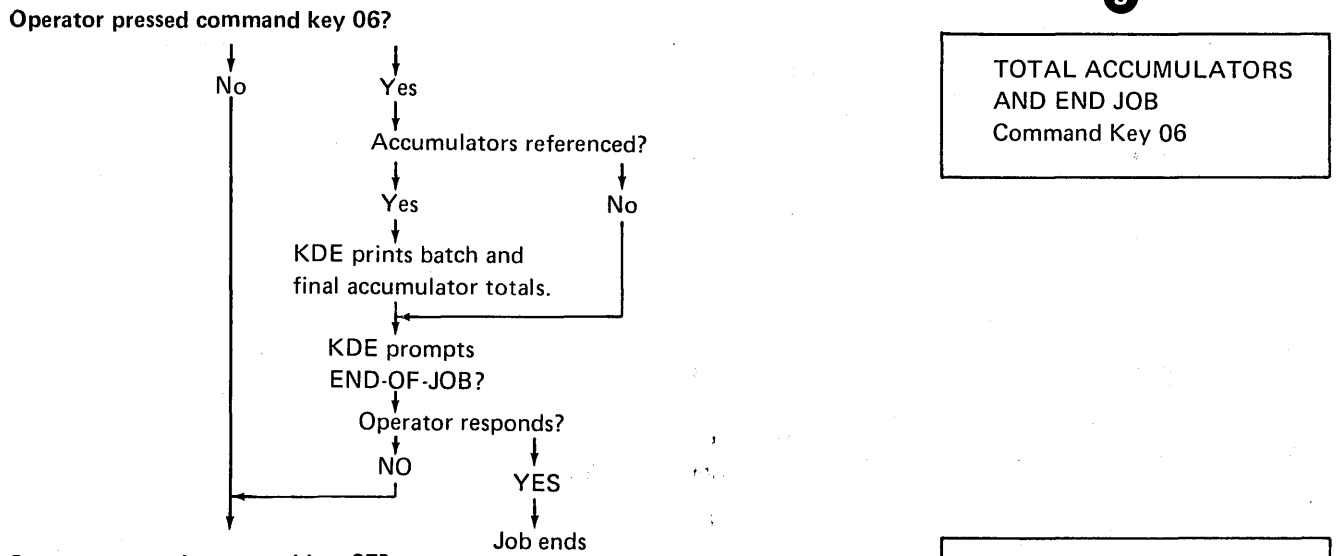
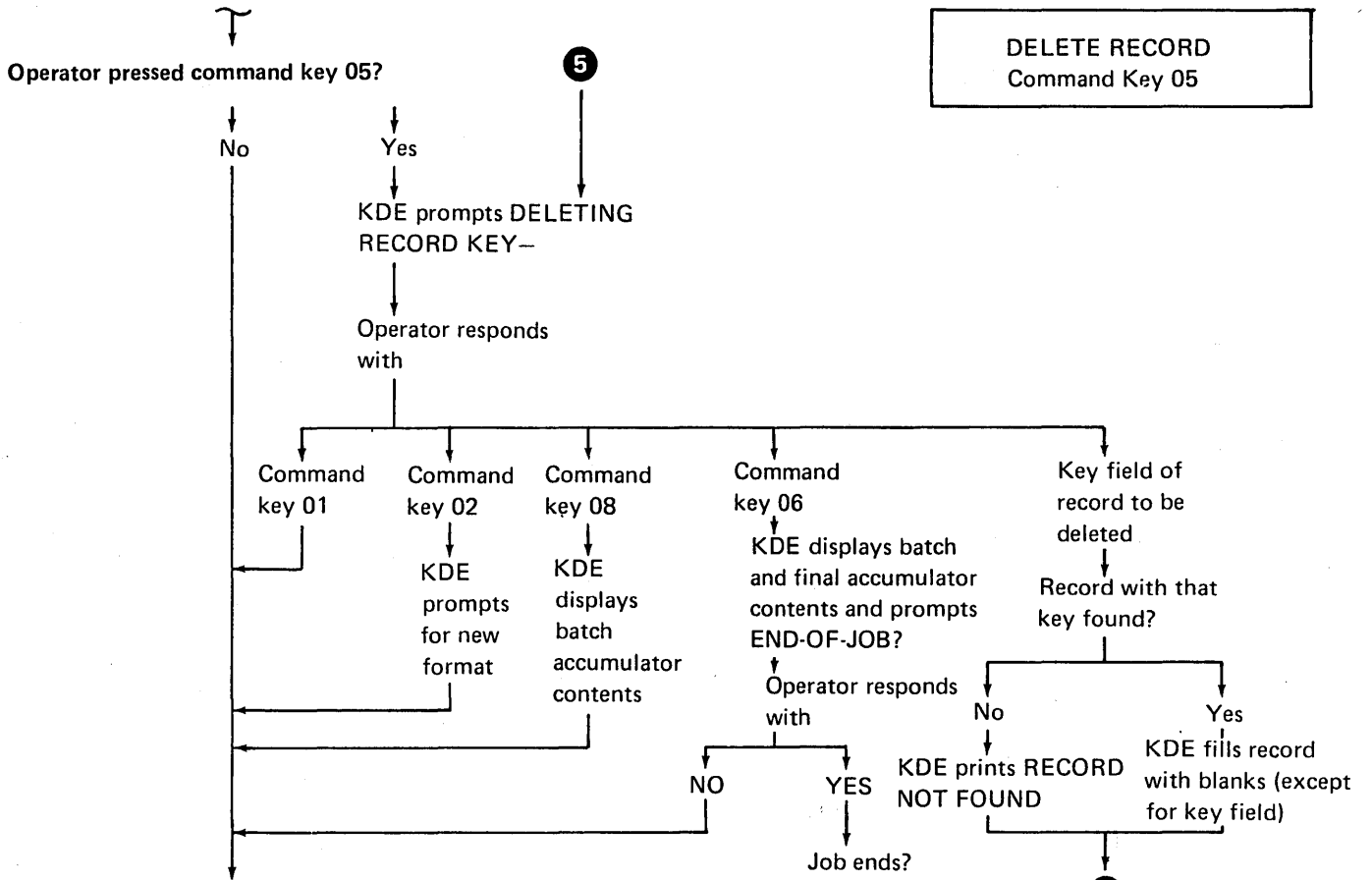
2

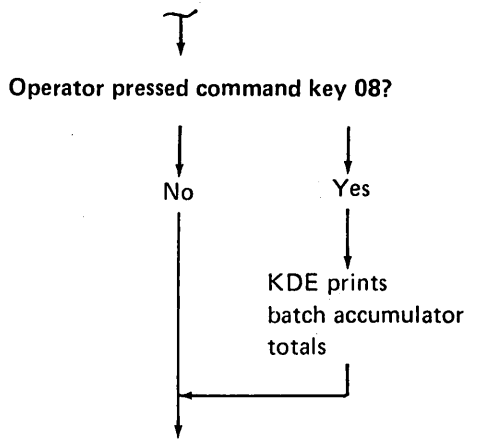
CORRECT PREVIOUS RECORD
Command Key 01

Operator pressed command key 01?

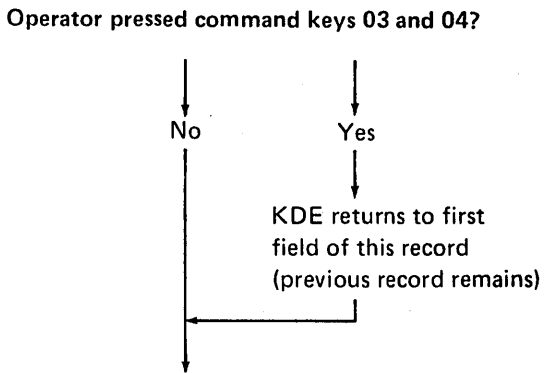




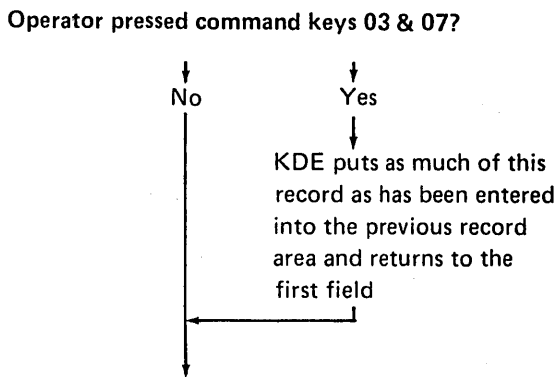




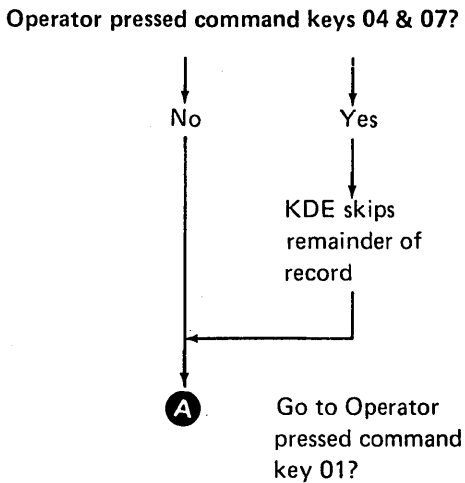
TOTAL BATCH
ACCUMULATORS
Command Key 08



REENTER CURRENT
RECORD
Command Keys 03 & 04



CORRECT PREVIOUS
FIELD IN CURRENT
RECORD
Command Keys 03 & 07



SKIP REMAINDER
OF RECORD
Command Keys 04 & 07

INCLUDING KDE CONTROL STATEMENTS IN PROCEDURES

The KDE program is frequently used for entering the same type of data. Rather than go through the keyword-response sequence each time, the operator can include the information needed in control statements and include these control statements in an OCL procedure (OCL BUILD cycle). The procedure can then be called (OCL CALL cycle) by the operator, and KDE will get the necessary information from the control statements. Delayed responses (?) are allowed in the control statements. When you use chained procedures (OCL BUILD cycle) or press the enter minus key after responding to CALL NAME, the system will not prompt MODIFY after the OCL statements and again after the including statements, thus further reducing operator intervention. The BUILD, BUILD, and CALL cycles are described in the *IBM System/3 Model 6 Operation Control Language and Disk Utility Programs Reference Manual*, GC21-7516. Examples showing KDE control statements in procedures are included at the end of this chapter.

Entering Control Statements

Every control statement consists of an identifier and parameters. The identifier is the first word of the statement and for KDE is always //KDE in positions 1-7. Parameters are information you supply to the program. Every parameter consists of a keyword, which identifies the parameter, followed by the response, which you supply.

Coding Rules

The rules for writing control statements are as follows:

1. *Statement Identifier.* All control statements must have //KDE in positions 1-7.
2. *Statement Parameters.* Parameters can be in any order. Use a comma to separate one parameter from another. Use a hyphen (-) within each parameter to separate the keyword from the response. Do not use blanks within or between parameters.
3. *Statement Length.* Control statements must not exceed 96 characters. Each control statement may contain one or more parameters. (See following example.) Do not use a comma at the end of the control statement. A blank must follow the last response and indicates end of statement.

1	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64
//	KDE	FNAME-	FORMKD,	FUNIT-	RI,	DSPLY-	NO,	NEW-	YES,	KSTART-	NO,	FNUM-	OI			
//	KDE	END														

OR

1	4	8	12	16	20	24	28	32
//	KDE	FNAME-	FORMKD,	FUNIT-	RI			
//	KDE	DSPLY-	NO,	NEW-	YES			
//	KDE	KSTART-	NO,	FNUM-	OI			
//	KDE	END						

End Control Statement

The END statement is a special statement that indicates the end of control statements. It must contain `//KDEEND` in positions 1-10 and must always be the last control statement for the program.

Control Statement Diagnostics

The control statements are diagnosed by KDE as they are read from the procedure. Keywords having delayed responses (?) are prompted in conversational mode. If required parameters are not in the control statements, a message is printed and the keywords are prompted in conversational mode. If errors are detected in the control statements, an asterisk is printed under the error parameter, an error message is printed, and conversational mode is entered. The control statements are not updated by the information supplied in conversational mode.

Control Statement Summary Chart

The following chart shows the keywords and responses for the parameters in the KDE control statements. In addition to the indicated responses, the question mark (?) may be entered to specify a delayed response.

Keyword	Response	Considerations
FNAME	1-6 character name	You supply KDE with the name of the source library entry that contains the KDE format descriptions.
	P/S	You will not use format descriptions. All records will be free-form. This response is equivalent to pressing the program start key as the response to the FORMAT NAME prompt in conversational mode.
FUNIT	F1, F2, R1 or R2	You supply KDE with the disk unit number of the source library entry that contains the KDE format descriptions.
DSPLY	YES or NO	You respond YES if you want KDE to display the format descriptions. You respond NO to bypass the display.
NEW	YES	You are creating a new KDE file.
	NO	You are going to process an existing KDE file.
KSTRT	nnn	You furnish KDE with the starting location of the key field.
	NO	KDE will use the first five bytes of each record as the key field, and enter a number starting with 00000 and incremented by 10 for each record. KLEN is not required.
	NO,P	KDE will use the first three bytes of each record as the key field, and enter a packed decimal number starting with 00000 and incremented by 10 for each record.
KLEN	nn	Length of key field.
FNUM	nn	You tell KDE which format description you want to use (nn is the 2-digit number in the header statement).

OCL LOAD CYCLE FOR KDE

Prompt	Response	Considerations
READY	LOAD	You wish to load a program (KDE) for execution.
LOAD NAME	\$KDE	Provide the name of the program to be executed.
UNIT	R1, R2, F1, or F2	Location of the disk in whose object library \$KDE resides.
DATE	Program start key	The system date is used.
SWITCH	Program start key	The KDE program does not use the switch indicators.
FILE NAME	KDEFILE	The KDE program requires the use of this file name for identification purposes.
UNIT	R1, R2, F1, or F2	Location of the disk file.
PACK	Disk name	Name assigned to disk during initialization.
LABEL	1-8 character name	Name of the data file you are creating or maintaining. (PAYROL1, for example.) This is the VTOC name on disk.
RECORDS	1-999999	If you are creating a data file, you must supply its size.
LOCATION	Program start key	Reference to file location is optional.
RETAIN	P, T, S, or A	File designation.
DATE	mm/dd/yy or dd/mm/yy	If you are maintaining an existing data file, supply its creation data if another KDEFILE with the same name exists on this pack.
FILE NAME	Program start key	No other files are used.
MODIFY	RUN	Begin execution of the KDE program.

KDE ERROR MESSAGES

Message	Meaning
BATCH ACCUMULATOR nn OVERFLOWED or FINAL ACCUMULATOR nn OVERFLOWED	The accumulator specified by nn has overflowed. The total before it overflowed will be printed and the accumulator set to zero.
DUPLICATE KEY	You have entered a record with the same key field as a previous record. You must re-enter the record with a different key field or use command keys.
FORMATS NOT FOUND	The module in the source library for your KDE format descriptions was not found. If inquiry was taken, KDE could not find the formats when it was brought back after inquiry. KDE will reprompt FORMAT NAME and FORMAT UNIT.
ERRORS DETECTED – CONVERSATIONAL MODE ENTERED	<p>While reading KDE control statements from a cataloged procedure, one or more of the following errors was detected.</p> <ol style="list-style-type: none"> 1. <i>Duplicate Keywords.</i> A keyword appears more than once in the control statements. 2. <i>Punctuation.</i> The format of the control statements is wrong. An extra or missing comma is the most common type of error. A blank in or between parameters (except following a comma) causes all characters following to be ignored. 3. <i>Keyword Spelling.</i> A keyword is spelled incorrectly. 4. <i>Response too Long.</i> The response to a keyword contains more characters than allowed. 5. <i>Invalid Identifier.</i> The control statement does not start with //bKDEb in positions 1-7. 6. <i>Invalid Response.</i> The response given is invalid for this keyword. <p>An asterisk (*) is printed under the first position of the keyword or response in error.</p>
INCONSISTENT KEY LENGTH	The key field length supplied does not agree with the key length of the KDEFIL the user is attempting to maintain.
INCONSISTENT KEY START	The key field start supplied does not agree with the key field start of the KDEFIL the user is attempting to maintain.
INCONSISTENT RECORD LENGTH	The record length supplied does not agree with the record length of the KDEFIL the user is attempting to maintain.
INVALID FORMAT NAME	The name you keyed as a response to the prompt FORMAT NAME was invalid. KDE re-displays the prompt after printing the error message.

Message	Meaning
INVALID FORMAT NUMBER n	<p>n = 0 Your response to SELECT FORMAT NUMBER was not a valid 2-digit number.</p> <p>1 Header record requested was not found.</p> <p>2 Free-form was requested and record length exceeds page width.</p>
INVALID FORMAT UNIT	<p>The disk unit number you keyed as a response to the prompt FORMAT UNIT was invalid. F1, F2, R1, or R2 are the only valid responses. F2 or R2 may also be invalid responses if you do not have a second disk drive on your system. KDE re-displays the prompt after printing the error message.</p>
INVALID KEY FIELD n	<p>n=0 The response to KEY FIELD START was zero, contained non-digits, or was over three digits.</p> <p>1 The response to KEY FIELD START exceeds record length.</p> <p>2 The response to KEY FIELD LENGTH was zero, contained non-digits, or was over three digits</p> <p>3 The response to KEY FIELD LENGTH was greater than 29.</p> <p>4 The sum of responses to KEY FIELD START and KEY FIELD LENGTH exceeds the record length.</p>
INVALID SELF-CHECK DIGIT	<p>The digit entered as the units position of a self-check field was not valid.</p>
KDE END OF JOB	<p>This message is printed when you press command key 06 before KDEFILE is opened and then respond YES to the prompt END-OF-JOB.</p>
KEYWORD REQUIRED BUT NOT GIVEN – PROCESSING CONTINUED	<p>A KDE control statement in a cataloged procedure does not contain a required keyword. The missing keyword is prompted in conversational mode.</p>
KEYWORD WAS GIVEN BUT NOT USED – PROCESSING CONTINUED	<p>One or more of the following keywords were included in a cataloged procedure but were not needed: FUNIT, DSPLY, KLEN, FNUM.</p> <p>The keywords are ignored and processing continues. Example: KSTRT-NO, KLEN-05.</p>
MINUS DATA IN KEY FIELD	<p>You have entered minus data into a numeric field that is part of the key field. The record cannot be maintained. Re-enter data as positive number.</p>
MISSING KEY FIELD	<p>You have requested that KDE supply the key fields but the format you have selected does not have a valid first field. (A005 or N005 for normal key or P003 for packed key.)</p>
MISSING END STATEMENT	<p>A cataloged procedure containing KDE control statements does not have an END control statement.</p>

Message	Meaning
PACKED FIELD IN KEY FIELD	The format descriptions you have selected contain a packed field as part of the key field such that records entered could not be maintained. Select a different format number.
RECORD KEYS WILL BE SORTED. WAIT UNTIL END OF JOB IS INDICATED.	This message is printed when you press command key 06 and then respond YES to the prompt END-OF-JOB. The job is not complete until KDE has sorted the keys in the indexed file. The job is complete when the system halt lights indicate end-of-job or the system prompts READY.
RECORD NOT FOUND	A record with the key field you have requested cannot be found in the file. KDE will re-prompt for the record key.
TOO MANY COMMAND KEYS	You have pressed more command keys than are allowed.
USER MUST SUPPLY KEY	KDE is supplying the keys for this file and has used keys up to 99990. KDE can increment no further so operator must supply keys for any additional records in this file.
YOU CANNOT CHANGE KEY	You have corrected a previously entered record and have changed the key field. Because KDE files are indexed using the key field, the key field cannot be changed during maintenance. KDE will re-prompt correcting record key.
YOU HAVE FILLED FILE	There is no more room in the file. The last record entered will not be in file. Command keys 01, 02, 05, 06, and 08 are allowed.

KDE SAMPLE JOBS

KDE Sample Job 1. Creating a New Keyboard Data Entry File

```
READY-                                LOAD
*****
010 LOAD          NAME-                $KDE
011              UNIT-                F1
020 DATE (12/31/23) -
030 SWITCH (00000000) -
040 FILE          NAME-                KDEFILE
041              UNIT-                R1
042              PACK-                R1R1R1
043              LABEL-               MASTER
044              RECORDS-
045              TRACKS-               2
046              LOCATION-
047              RETAIN-               P
048              DATE-
050 FILE          NAME-
*****
```

MODIFY

RUN

FORMAT NAME - FORM01

FORMAT UNIT - R1

DISPLAY FORMATS ? YES

H01065

N005

A020

A020

N005A1

N005A2

A010

NEW KDE FILE ? YES

KEY FIELD START - NO

SELECT FORMAT NUMBER - 01

* * * * *

00000 CITY SUPPLY CO. PORTLAND, ORE. 50000 5000 888-763-65

00010 AJAX HARDWARE DAYTON, OHIO 3000 430 545-672-82

00020 A-1 PARTS SUPPLY MIAMI, FLA. 45000 2090 326-951-90

00030

(Press command key 08)

KDE Sample Job 2. Maintaining a Keyboard Data Entry File

This job will correct a KDE record that was entered in sample job 1.

```

READY-                                LOAD
*****
010 LOAD          NAME-                SKDE
011              UNIT-                F1
020 DATE (12/31/23) -
030 SWITCH (00000000) -
040 FILE          NAME-                KDEFILE
041              UNIT-                R1
042              PACK-                R1R1R1
043              LABEL-               MASTER
044              RECORDS-
045              TRACKS-                2
046              LOCATION-
047              RETAIN-                P
048              DATE-
050 FILE          NAME-
*****
MODIFY

RUN

FORMAT NAME -                FORM01
FORMAT UNIT -                R1
DISPLAY FORMATS ?           YES

H01065

N005

A020

A020

N005A1

N005A2

A010

NEW KDE FILE ?              NO
KEY FIELD START -          NO
SELECT FORMAT NUMBER -      01
* * * * *
00030                       (Press command key 01)
CORRECTING RECORD KEY -     70
RECORD NOT FOUND
CORRECTING RECORD KEY -     10
00010 AJAX HARDWARE        DAYTON, OHIO        03000 00430 545-672-82
00010 (Press command key 04) AJAX HARDWARE VILLISCA, IOWA (Press command key 07) 03000 00430 545-672-82
CORRECTING RECORD KEY -     (Press command key 06)

```

BATCH ACCUMULATORS	00	01	02	03	04
	0	98000	7520	0	0
	05	06	07	-08	09
	0	0	0	0	0
FINAL ACCUMULATORS	00	01	02	03	04
	0	98000	7520	0	0
	05	06	07	08	09
	0	0	0	0	0

END OF JOB ? YES

RECORD KEYS WILL BE SORTED. WAIT UNTIL END OF JOB IS INDICATED.

KDE Sample Job 3. Building a KDE Procedure to Create a New File

```
READY-
000 BUILD          NAME-          BUILD
001              UNIT-          TEST01
010 LOAD          NAME-          SKDE
011              UNIT-          F1
*****
020 DATE          -
030 SWITCH (00000000) -
040 FILE          NAME-          KDEFILE
041              UNIT-          R1
042              PACK-          R1R1R1
043              LABEL-         FILE01
044              RECORDS-
045              TRACKS-         2
046              LOCATION-
047              RETAIN-         P
048              DATE-
050 FILE          NAME-
*****
MODIFY

INCLUDE
*****
ENTER UTILITY CONTROL STATEMENTS
00

* CREATING A NEW FILE      (This is a comment. Comments can be included in control
01                        statements by entering an asterisk in position one.)

// KDE FNAME=KDFORM,FUNIT=R1,DSPLY=YES,NEW=YES,KSTRT=NO,FNUM=01
02

// KDE END
03

RUN
*****
MODIFY

RUN
```

**KDE Sample Job 4. Calling Procedure Built by Sample
Job 3.**

This job will create a KDE data file.

```
READY-          CALL
000 CALL        TEST01
001             UNIT-  R1 (Press ENTER-key to suppress printing of OCL procedure)
```

* CREATING A NEW FILE

// KDE FNAME-KDFORM,FUNIT-R1,DSPLY-YES,NEW-YES,KSTRT-NO,FNUM-01

// KDE END

H01050

N005

N005

A010

D020

R005

S005

* * * * *

00000 4732 XY COMPANY 03/27/72 3140

00010 2480 XZ COMPANY 03/27/72

00020 (Press command key 06)

END OF JOB ? YES

RECORD KEYS WILL BE SORTED. WAIT UNTIL END OF JOB IS INDICATED.

KDE Sample Job 5. Calling a KDE Procedure to Maintain the File Built by Sample Job 4.

```

READY-
000 CALL          CALL
001              NAME- TEST02 (P/S)
                UNIT-  R1 (P/S)
*****
010 LOAD          NAME-SKDE
011              UNIT-F1
020 FILE          NAME-KDEFIL.E
021              UNIT-R1
022              PACK-R1R1R1
023              LABEL-FILE01
024              TRACKS-2
025              RETAIN-P

```

Because the operator did not press the ENTER key after entering TEST02 or R1, the procedure is displayed and the operator must enter RUN after both MODIFY prompts.

```

*****
MODIFY

```

```

RUN ←

```

```

*****
INCLUDED STATEMENTS
00 * MAINTAINING A FILE
01 // KDE FNAME-KDFORM,FUNIT-R1,KSTRT-NO,FNUM-01,NEW-NO,DSPLY-NO
02 // KDE END

```

```

*****
MODIFY

```

```

RUN ←

```

* MAINTAINING A FILE

// KDE FNAME-KDFORM,FUNIT-R1,KSTRT-NO,FNUM-01,NEW-NO,DSPLY-NO

// KDE END

* * * * *

00020 7693 ABC INC. 03/28/72 5400

00030 8427 XX COMPANY 03/28/72 927

00040

CORRECTING RECORD KEY - 10

00010 02480 XZ COMPANY 03/27/72 (Press command key 04 for both fields)

00010 02480 XZ COMPANY 03/27/72 132

CORRECTING RECORD KEY - (Press command key 08)

END OF JOB ? YES

RECORD KEYS WILL BE SORTED. WAIT UNTIL END OF JOB IS INDICATED.

**KDE Sample Job 8. Calling a Procedure Containing Errors
in KDE Control Statements**

```

READY-          CALL
000 CALL          TEST05
001              UNIT- R1 (Press ENTER-key)

// KDE FNAME-KDEFORM,FUNIT-T1,DSPLYNO
           * 1 *          * 2 * 3 *
//KDE FNUM-01,NEW-YESKSTRT-1,KLEM-5
* 4 *   * 5 *           * 6 *   * 7 *
// KDE END

*** ERRORS DETECTED--CONVERSATIONAL MODE ENTERED ***

FORMAT NAME -          KIFORM
FORMAT UNIT -          R1
DISPLAY FORMATS ?     NO
NEW KDE FILE ?        YES
KEY FIELD START -     1
KEY FIELD LENGTH -    5
SELECT FORMAT NUMBER - 01
* * * * *
00000 487 JKL INC. 03/30/72          98
(Press command key 02)

SELECT FORMAT NUMBER - 00
00010 THIS IS FREE FORM.
(Press command key 06)

END OF JOB ?          YES
RECORD KEYS WILL BE SORTED. WAIT UNTIL END OF JOB IS INDICATED.

```

- 1 Response too long. Only six characters are allowed in the format name.
- 2 Invalid response. T1 is not a valid unit.
- 3 Keyword spelling. Because the (-) is missing between DSPLY and NO, KDE recognized DSPLYNO as an invalid keyword.
- 4 Invalid identifier. The blank between // and KDE is missing.
- 5 Position 8 is flagged because a keyword does not start there. The first keyword must start in position 8. Error 4 caused this error.
- 6 Invalid response. Because the comma is missing between parameters, YESKSTRT is flagged as an invalid response to the keyword NEW.
- 7 Keyword spelling. KLEM should be KLEN.

Chapter 4. Data Interchange Utility (DIU)

The Data Interchange Utility program (DIU) converts disk data files created in System/3 BASIC format to the format used by the disk system management programs. Disk system management data files can also be converted so that they can run on System/3 BASIC. This conversion allows the co-resident user (systems with both programming systems) to gain full use of the computing power of the System/3 BASIC and the flexibility and report writing capability of the disk system management programs.

Conversion is necessary because the data files for each system have a different format. The System/3 BASIC data files are made up of field strings, whereas the disk system management data files are made up of records. DIU conversion results in a newly created data file in the new format.

The DIU program can create a disk system management data file when converting from System/3 BASIC. However, when converting to System/3 BASIC, the System/3 BASIC data file must have been previously defined.

The disk system management file must have DIUFILE as the file name.

The DIU program operates under control of the disk system management using normal OCL sequences. The OCL must describe the disk system management data file. Information about the System/3 BASIC data file and fields within the disk system management records are described by DIU control statements. The possible ways of supplying OCL and DIU control statements are shown later in this chapter.

DATA FILE CONVERSION

Any System/3 BASIC data file can be converted to a consecutive (sequential) data file. That is, the input can be from any of the three types of libraries supported by System/3 BASIC: pooled, two-star, and user. The System/3 BASIC file name supplied to the DIU program identifies the library type:

- One asterisk (*) at the beginning of file name — Pooled library
- Two asterisks (**) at the beginning of file name — Two-star library
- No asterisks at beginning of file name — User (password) library

Indexed or sequential disk system management data files can be converted to System/3 BASIC format. However, before the System/3 BASIC file can be written, System/3 BASIC must allocate space for the file. The System/3 BASIC file is written as a program generated file in the user or two-star library. The maximum record length of a disk system management data file to be converted is 256 positions with a maximum of 99 fields per record. All fields of the data file need not be converted into the System/3 BASIC data file format. That is, only the fields required by the System/3 BASIC programs that are to use this data need be described. This is possible because System/3 BASIC data files are created as strings of fields rather than records. Fields that are not described are not included in the System/3 BASIC file.

System/3 BASIC Data Fields

Three types of fields are allowed in a System/3 BASIC data file: alphameric, short precision floating point numeric, and long precision floating point numeric. Each type of field has a fixed length and contains the data and status information about the field.

Alphameric Fields: This field type has 18 positions available for data and one position of status, making a total of 19 positions.

Short Precision Numeric Fields: This type of field is five positions in total length, with space for seven numbers, an exponent, and a sign (plus or minus).

Long Precision Numeric Fields: This type of field is nine positions in total length, with space for 15 numbers, an exponent, and a sign (plus or minus).

Disk System Management Data Fields

There are four types of data fields in a disk system management data file, three of which are used by the DIU program: alphameric, packed numeric, and unpacked numeric. (Binary fields cannot be converted.)

Alphameric Fields: Alphameric fields are variable in length with a maximum of 256 positions (the maximum record length in a data file to be converted to System/3 BASIC is 256 positions). The total field length is available for data.

Packed Numeric Fields: A packed numeric field is variable in length with a maximum length of eight positions. Each position except the units position contains two numbers. The units position contains one number and the sign (plus or minus). The maximum field length (8) can therefore contain 15 numbers and a sign. The maximum number of decimal places is nine for any data field.

Unpacked Numeric Fields: An unpacked numeric field is variable in length with a maximum of 15 positions. Each position contains one number. The units position contains, in addition to the number, the sign of the field (plus or minus). The maximum number of decimal places that may be specified is nine.

DESCRIBING DISK SYSTEM MANAGEMENT FIELDS

The field type, starting location, and ending location of the fields within the disk system management data file record must be described before the DIU utility can process the file. Only those fields which are to be converted must be described. The starting and ending locations can be anywhere within the record as long as the described fields do not overlap and the starting location is not greater than the ending location.

Each field in the disk system management record to be created or converted must be described using the following DIU field description:

nnntlll (alphameric fields)
or
nnntlll.d (packed or unpacked numeric fields)

Where:

nnn is the starting location of the field
t is the field type. Valid field types are:

- A – Alphameric
- N – Unpacked numeric. The field must contain all valid digits (0-9). No blanks are allowed.
- P – Packed numeric

lll is the ending location of the field
.d is the number of decimal places in a packed or unpacked numeric field.

The maximum number of DIU field descriptions for a disk system management record is 99.

DISK SYSTEM MANAGEMENT FIELD DESCRIPTION DIAGNOSTICS

The disk system management file field descriptions are diagnosed for the following errors:

1. Starting location of field not within record.
2. Ending location of field not within record.
3. Starting location numerically greater than ending location.
4. Field type not A, N, or P.
5. Number of decimal places (d) not provided for an N or P type field.
6. Number of decimal places greater than 9 or not numeric.
7. Current DIU field description overlaps part of a previously described field.
8. Unpacked numeric field length greater than 15.
9. Packed numeric field length greater than 8.
10. Number of decimal places exceeds field length.

CONSIDERATIONS FOR CONVERTING ALPHAMERIC

When converting alphameric fields using DIU, the relationship of the two field lengths determines the position of the data within the field.

CONVERTING FROM DISK SYSTEM MANAGEMENT TO SYSTEM/3 BASIC

Field Length Relationship	Result
System/3 BASIC field longer	Left-justify and fill remaining positions with blanks
Field lengths equal	One-for-one conversion
Disk system management field longer	Create enough System/3 BASIC fields to hold the disk system management field. (If the last System/3 BASIC field has less than 18 characters, the field will be left-justified and the remaining positions filled with blanks.)

CONVERTING FROM SYSTEM/3 BASIC TO DISK SYSTEM MANAGEMENT

Field Length Relationship	Result
System/3 BASIC field longer	The characters in the rightmost positions are dropped
Field lengths equal	One-for-one-conversion
Disk system management field longer	Left-justify and fill remaining positions with blanks

CONSIDERATIONS FOR CONVERTING NUMERIC FIELDS FROM SYSTEM/3 BASIC

In converting numeric fields, consideration must be given to the maximum value a given size field will accommodate. If this maximum is exceeded, a loss of significant digits occurs.

DEFINITIONS THAT CAUSE MOST SIGNIFICANT DIGIT TO BE DROPPED

System/3 BASIC Field	Field Definitions
Field with negative or zero exponent	Value of exponent greater than number of decimal positions in disk system management field*
Field with positive exponent	Value of exponent greater than number of digits in whole number portion of disk system management field*
*The system halts. The operator can then override the halt and continue the conversion. If he does, the numeric value of the resulting disk system management field will be zero.	

DIU COMMAND KEY CHART

Command Key	Function	Considerations
01	Correct a field description	When the operator wants to change a previously keyed DIU field description, he presses command key 01. DSM FIELD NUMBER is then prompted and the operator provides the statement number of the DIU field description to be changed. The statement number is prompted on the next line and the operator keys in the changed field description. This description then replaces the old one.
	Delete a field description	To delete a previously entered field description, after pressing command key 01, the operator responds to DSM FIELD NUMBER with the statement number of the DIU field description to be deleted. The statement number is then prompted on the next line. The operator then enters DELETE. The field description is deleted.
06	End the job	The operator is prompted END OF JOB. If the operator responds YES, the job ends before file conversion begins. If the operator responds NO, the DIU utility proceeds with the next prompt.

Note: The other command keys cannot be used.

METHODS OF SUPPLYING OCL AND DIU STATEMENTS

Possibilities	How Operator Supplies the OCL Statements	How Operator Supplies the DIU Control Statements
Standard Prompting	Printer and Keyboard	Printer and Keyboard
Cards	On Cards (OCL must be in Disk System format)	On Cards (see <i>DIU Control Card Format and Sequence</i>)
Procedure containing DIU Control Statements	Printer and Keyboard 1. Use the OCL BUILD cycle to build a procedure of the OCL statements. Use the keyword INCLUDE to add the DIU control statements (maximum 25) to the procedure. 2. Use the OCL CALL cycle to run the DIU program.	No operator action required. DIU control statements are part of the procedure. DIU control statements included in a procedure must be in card format. (See <i>DIU Control Card Format and Sequence</i>)
Standard Prompting for OCL. DIU Statements On Cards	Printer and Keyboard	On cards (see <i>DIU Control Card Format and Sequence</i>)

USING THE KEYBOARD TO SUPPLY THE DIU CONTROL STATEMENTS

The operator keys in responses to DIU supplied prompts to describe the System/3 BASIC file and the fields of disk system management records.

Each response is checked for errors. If errors are found, an error message is printed and the prompt is redisplayed. The operator may then key in a new response or press command key 06 to end the job (see *DIU Command Key Summary Chart* in this chapter).

DIU KEYWORD RESPONSE SUMMARY CHART

Keyword	Responses	Considerations
BASIC FILE	Name, *name, or **name	Must be an existing file. Maximum file name length is 10 characters (with**).
	CARD INPUT	All DIU control statements will be read from cards. (See Appendix E for DIU control statement card formats.)
	Command Key 6	DIU will prompt for END-OF-JOB
BASIC PASSWORD —	Password of protected disk file	PASSWORD is prompted if response to FILENAME did not begin with a single or double asterisk (* or **). Maximum length of the protected disk file's password is 8 characters.

DIU KEYWORD RESPONSE SUMMARY CHART (continued)

Keyword	Responses	Considerations
BASIC FILE UNIT –	R1,R2,F1, or F2	Location of the disk unit which contains the file: <ul style="list-style-type: none"> ● R1 – Removable disk on drive one. ● R2 – Removable disk on drive two. ● F1 – Fixed disk on drive one. ● F2 – Fixed disk on drive two.
INPUT FILE TYPE –	BASIC	Convert from System/3 BASIC.
	DSM	Convert to System/3 BASIC.
DSM RECORD SIZE	nnn	Length of each record in the disk system management file. If convert from System/3 BASIC, operator must respond with record length. If convert to System/3 BASIC, record length in existing file is displayed.
DESCRIBE DSM FIELDS	No response	This prompt tells the operator that DIU is going to ask for the DIU field descriptions of the disk system management records.
000	nnntlll or nnntlll.d	See <i>Describing Disk System Management Fields</i> in this chapter.
nnn	nnntlll or nnntlll.d	The operator keys in the next DIU field description. The DIU field description is then checked for errors. A message is printed if an error exists, and the operator is reprompted for a correct description.
	Command Key 1	To change a previously keyed DIU field description.
	Command Key 6	The operator may end the job.
	END	Begin file conversion.
DSM FIELD NUMBER	nnn	When command key 01 is pressed, operator keys in the statement number of the DIU field description to be changed.
	Program Start key	DIU will reprompt field number prompted before command key 01 was pressed.
nnn	Operator keys in corrected DIU field description	The corrected DIU field description replaces previous DIU field description.
	DELETE	The DIU field description is deleted.
END OF JOB	YES	After pressing command key 06, a YES response ends the job.
	NO	The operator wants to continue the job. DIU will reprompt keyword prompted before command key 06 was pressed.

DIU CONTROL STATEMENTS ON CARDS

If the DIU control statements are entered via the data recorder the first 64 columns of the card will be printed. Invalid cards will be flagged with an error message. The DIU program will halt allowing the operator to repunch the card and continue.

Command keys 01 and 06 are not allowed when input is on cards.

When DIU is loaded from cards, reader control returns to the keyboard after the // RUN card is read, and BASIC FILE NAME is prompted on the printer. To read the rest of the procedure from cards, the response CARD INPUT must be typed. Reader control will then pass to the data recorder again, and the next card read should be:

// DIU BASIC FILE NAME –

DIU CONTROL CARD FORMAT AND SEQUENCE

Each DIU control statement must be on a separate card and must contain //DIU in columns 1-7. The following chart shows the entries for and the sequence of each control statement:

Entry	Considerations
BASIC FILE NAME-name	Must be an existing file (name, *name, or **name). Maximum file name length is 10 characters.
BASIC PASS- WORD-name	Required if file is in the password library. Maximum length is 8 characters. This card is not used if the file name begins with * or **.
BASIC FILE UNIT-nn nn is R1 F1 R2 F2	Location of the disk containing the file: <ul style="list-style-type: none"> ● R1 is removable pack on drive one. ● F1 is fixed pack on drive one. ● R2 is removable pack on drive two. ● F2 is fixed pack on drive two.
INPUT FILE TYPE BASIC or DSM	BASIC to convert from System/3 BASIC. DSM to convert to System/3 BASIC.
DSM RECORD SIZE – nnn nnn is record length	Maximum record length is 256 positions. This card is not used for conversion to System/3 BASIC.
FIELD-nnntIII or FIELD-nnntIII.d or FIELD-END	See <i>Describing Disk System Management Fields</i> in this chapter. One card is required for each field to be converted. The DIU field descriptions must be in the same sequence as the System/3 BASIC file. Maximum number of fields is 99.

DIU CONTROL STATEMENTS INCLUDED IN PROCEDURE

If the DIU control statements (in card format) are included in the OCL procedure the first 64 bytes of each statement will be printed. Invalid statements will be logged and END-OF-JOB called.

Command keys and retry options are not available when DIU control statements are included in the OCL procedure.

OCL LOAD CYCLE FOR DIU

Prompt	Response	Considerations
READY	LOAD	You wish to load a program (DIU) for execution.
LOAD NAME	\$DIU	Provide the name of the program to be executed.
UNIT	R1, R2, F1, or F2	Location of the disk in whose object library \$DIU resides.
DATE	Program Start key	The system date is used.
SWITCH	Program Start key	The DIU program does not use the switch indicators.
FILE NAME	DIUFILE	Refers to the disk system management file. File can be either: <ul style="list-style-type: none">• An existing file• A file that will be created during the DIU program run. (Maximum record length is 256.)
UNIT	R1, R2, F1, or F2	Location of the disk which contains the file.
PACK	Disk name	Name which identifies disk on which the file is located.
LABEL	VTOC file name	Name of data file being created or converted.
RECORDS	1-999999	If the data file is being created, you must supply its size.
LOCATION	Program Start key	Reference to file location is optional.
RETAIN	P, T, or S	File designation.
DATE	mm/dd/yy or dd/mm/yy	If you are converting an existing file, you must supply its creation date if another DIUFILE with the same label exists on this pack.
FILE NAME	Program Start key	No other files are used.
MODIFY	RUN	System is ready to begin prompting for DIU.

DIU ERROR MESSAGES

If an invalid response is keyed in while supplying DIU control statements, an error message is printed. The following chart lists the error messages and their meanings:

Message	Meaning
// DIU MISSING	Columns 1-6 of input from cards or procedure is not // DIU (see <i>DIU Control Card Format and Sequence</i>).
BASIC OPEN ERROR – XX	Where XX is: 03 – No BASIC library on unit specified. 04 – BASIC password not on unit specified. 05 – Filename is not in BASIC library. 06 – BASIC file is not a data file. 07 – BASIC keyboard generated file is not valid for DIU output. 08 – A pooled library BASIC file is not valid for DIU output. 09 – A protected non-program generated BASIC file is not valid for DIU output. 12 – Permanent disk I/O error occurred when opening BASIC file.
BASIC TO DSM CONVERSION ERROR XX	Where XX is: 11 – Permanent disk I/O error occurred while processing BASIC file. Job cancelled. BASIC file is closed. DSM file is closed and usable but will not contain all data. 41 – Permanent disk I/O error occurred while processing DSM file. Job cancelled. BASIC file is closed. DSM file is closed but incomplete. 42 – End of DSM file is reached before end of BASIC file. Halt B34. Option 2 – BASIC file closed. DSM file closed and usable but incomplete. Option 3 – BASIC file closed. DSM file not closed and not usable.

Message	Meaning
	<p>43 – A numeric BASIC field is to be converted to an alphameric DSM field. Halt B235.</p> <p>Option 1 – DSM field will be blank.</p> <p>Option 3 – BASIC and DSM files are not closed. DSM output is not usable.</p>
	<p>44 – An alphameric BASIC field is to be converted to a numeric DSM field. Halt B235.</p> <p>Option 1 – Output field will be decimal zeros if a numeric DSM field, hexadecimal zeros if a packed DSM field.</p> <p>Option 3 – BASIC and DSM files are not closed. DSM output is not usable.</p>
	<p>45 – Negative exponent of BASIC field requires more decimal places than provided in DSM field. Same halt and options as error 44.</p>
	<p>46 – Positive exponent of BASIC field requires more whole number places than provided in DSM field. Same halt and options as error 44.</p>
<p>CARD READ ERROR</p>	<p>A read error occurred while reading DIU statements from cards.</p>
<p>DATA96 NOT ON LINE</p>	<p>The response to BASIC FILE NAME - was CARD INPUT but the card reader is not online.</p>
<p>DISK READ ERROR</p>	<p>Error occurred while reading DIU statements from procedure on disk.</p>
<p>DIUFILE IS NOT INDEXED OR SEQUENTIAL</p>	<p>Invalid file type. Job cancelled. DSM file has not been opened. BASIC file is closed with no data converted. DSM file is not usable as DIUFILE.</p>

Message	Meaning
DSM FIELD ERROR XX	<p>Where XX is:</p> <ul style="list-style-type: none"> 01 – More than 99 DSM fields. 02 – Field ending position less than field starting position. 03 – Invalid field type. Valid field types are A, N, and P. 04 – Number of decimal places not given for an N or P type field. 05 – Number of decimal places not 0-9. 06 – Current field overlaps previous field. 07 – Starting or ending position of field greater than record length. 08 – Numeric field length greater than 15. 09 – Packed field length greater than 8. 10 – Field start or end not numeric value 001-256. 11 – Number of decimal places greater than field length. 12 – Constant FIELD- is missing from card or procedure input. 13 – Decimal specified for A-type field.
DSM FIELD NOT FOUND	<p>During command key 1 processing, response to DSM FIELD NUMBER was not:</p> <ul style="list-style-type: none"> 1. A 3-digit number. 2. The field number of a previously entered DSM field description.
DSM TO BASIC CONVERSION ERROR XX	<p>Where XX is:</p> <ul style="list-style-type: none"> 12 – Permanent disk I/O error occurred while processing BASIC file. The DSM and BASIC files are closed. BASIC file is usable. 50 – Permanent disk I/O error occurred while processing DSM file. DIUFILE and BASIC output file are closed. BASIC output file is usable but does not contain all DSM file data.

Message	Meaning
	<p>52 – End of BASIC output file occurred before end of DSM input file. Halt B34.</p> <p>Option 2 – DSM and BASIC files are closed. BASIC file is usable but does not contain all input data.</p> <p>Option 3 – DSM file is not closed. BASIC file is closed and usable but does not contain all input data.</p>
	<p>53 – Packed DSM field was invalid sign. Halt B235.</p> <p>Option 1 – BASIC output field will be a zero field with a zero exponent.</p> <p>Option 3 – DSM and BASIC files are not closed. BASIC file is not usable.</p>
	<p>54 – Numeric DSM field has invalid sign. Halt B235.</p> <p>Option 1 – BASIC output field will be a zero field with a zero exponent.</p> <p>Option 3 – DSM and BASIC files are not closed. BASIC file is not usable.</p>
END CARD MISSING	The END card is missing from the card input deck. (See <i>DIU Control Card Format and Sequence</i> .)
INVALID BASIC NAME	Name entered for BASIC file is invalid.
INVALID BASIC PASSWORD	Password entered for BASIC file is invalid.
INVALID BASIC UNIT	Unit is invalid or not on system.
INVALID DSM SIZE	Size of DSM file must be 001-256.
INVALID INPUT TYPE	Input type must be BASIC or DSM.
INVALID RESPONSE	During command key 01 processing, response to nnn was END. During non-key 01 processing, response to nnn was DELETE.
RESTART CARD DECK	When reading DIU statements from cards, a disk error occurred. The DIU program must be restarted.

DIU SAMPLE JOB

You have a System/3 BASIC file you want to use in one of your disk system management programs. Before you can use the file; you must convert from the System/3 BASIC format. The System/3 BASIC file name is I. It is a protected file with a password PLEASE. Presently it is on the removable disk on drive one. The Data Interchange Utility program is on the fixed disk on drive one. You are going to provide the necessary OCL statements through an OCL LOAD sequence. Here is how you would code the file conversion.

```
READY-                                LOAD
*****
010 LOAD          NAME-                $TIU
011              UNIT-                F1
-----
020 DATE (11/11/11) -
030 SWITCH (00000000) -
040 FILE          NAME-                DIUFILE
041              UNIT-                R1
042              PACK-                D4ZB11
043              LABEL-               NEWDES
044              RECORDS-              400
045              LOCATION-
046              RETAIN-                T
047              DATE-
050 FILE          NAME-
*****
MODIFY

RUN

BASIC FILE NAME-                I
BASIC PASSWORD-                 PLEASE
BASIC FILE UNIT-                 R1
INPUT FILE TYPE-                 BASIC
DSM RECORD SIZE-                 100

DESCRIBE DSM FIELDS

000                                007N013.1
010                                015N021.2
020                                025N031.3
030                                041N047.1
040                                033N039.0
050                                060P064.3
060                                080A090
070                                (Press command key 01)

DSM FIELD NUMBER-                010
010                                015N020.2
070                                END
```

\$DIU 83
 \$KDE 62
 \$KSE 32
 *
 to indicate start of field
 KDE 49
 KSE 10
 start of DIU BASIC filename 77
 *name 77, 80
 (see also **name,name)
 **name 77, 80
 (see also *name,name)
 ,P 25
 [An]
 (see accumulators)

A
 (see alphameric fields)
 accumulators 52
 reference in KDE formats 9
 totals 52
 (see also command key 06)
 alphameric fields
 in format descriptions 7, 8
 DIU 77
 KDE 50
 KSE 10
 An
 (see accumulators)
 asterisk
 (see *)
 automatic duplication fields
 in format descriptions
 KDE 8
 KSE 7
 in KDE 50
 in KSE 10
 automatic duplication indicator
 changing condition 2
 (see also command key 03)
 effect of changing formats 10, 50
 indicating condition 2, 10, 50
 automatic skip fields
 in KDE 50
 in KSE 10

BASIC
 (see System/3 BASIC)
 batch accumulators
 (see accumulators)
 BKSP (backspace) key 2
 blanks
 deleting KDE record 54
 entering as response 1
 in alphameric fields 10, 50
 in source module name 5

C
 (see self-check fields)
 CANCEL 9
 CARD INPUT 80, 82
 cataloged control statements
 (see control statements)
 command keys 2, 11, 53, 79
 restrictions 9, 11
 (see also keyword-response summary charts)
 command key usage charts
 DIU 79
 KDE 53
 KSE 11
 control statements
 DIU 80, 82
 KDE 59
 KSE 26
 prompt-response 1
 conversion-restriction—DIU 78
 converting alphameric fields—DIU 78
 converting numeric fields—DIU 79
 co-resident programming system 76
 correcting current record 53
 correcting previous field 53
 correcting previous record 54
 CORRECTING RECORD KEY 54, 55
 CORRECTING STATEMENT NUMBER 11, 13
 correction cycle 47
 creating a new KDE file 49, 66
 creating format descriptions of KDE records 7, 20
 creating format descriptions of KSE statements 5, 18, 33
 creating formatted source statements 9, 22, 34
 creating free-form source statements 11

D
 (see automatic duplication fields)
 dd (format description number) 5, 8
 data records 50
 data files—KDE 47
 data file conversion 76
 date—KDEFIELD 47
 deleting KDE record 54
 deleting KSE statement 12
 DELETING RECORD KEY 54, 57
 DELETING STATEMENT NUMBER 12, 14
 describing disk system management fields 77
 diagnostics
 DIU 78
 format description
 KDE 9
 KSE 7
 KSE end-of-job 9
 disk name
 DIU OCL response 83
 KDE OCL response 62
 disk system management
 discussion 2, 76
 DIU response 80

disk system management data fields
 describing 77
 types
 alphanumeric 77
 packed numeric 77
 unpacked numeric 77
 disk system management files 77
 disk system record length 77
 DISPLAY FORMATS
 discussion 9
 KDE prompt 47
 KSE prompt 9
 DIU 2, 76
 DIU command key chart 79
 DIU control card format and sequence 82
 DIU control statements 82
 (*see also* DIU control card format and sequence; DIU
 keyword response summary chart)
 DIU error messages 84
 DIUFILE 83
 DIU keyword-response summary chart 80
 DIU OCL LOAD sequence 83
 DSM
 (*see* disk system management)
 duplication fields, automatic
 (*see* automatic duplication fields)
 duplication indicator, automatic
 (*see* automatic duplication indicator)
 duplicating fields
 Tab key 3
 (*see also* command key 04 and command key 07)
 duplicating from previous statement 6, 11, 14, 16

 END-OF-JOB
 KDE 57
 KSE 4, 15
 end-of-job diagnostics 9
 enter minus key
 when entering KDE records 2, 50
 enter plus key 2
 entering data records 50
 errors
 (*see* format description diagnostics)
 error messages
 DIU 84
 KDE 63
 KSE 28
 errors
 (*see* format description diagnostics)

 field descriptions statement
 format of 7
 part of format description 2
 field erase key 2
 field length
 DIU
 (*see* disk data management fields)
 KDE 7
 KSE 5
 field lights 2, 11, 53
 field start position
 (*see* *)

field types
 DIU 77
 KDE 7
 KSE 5
 files, KSE data 47
 form skip key 3
 format descriptions
 (*see also* free-form)
 of KDE data records 7, 20, 47
 (*see also* keyword-response summary charts)
 of KSE source statements 5, 18
 (*see also* keyword-response summary charts)
 FORMAT DESCRIPTION
 (*see* keyword-response summary charts)
 format description command key restrictions 9
 format description diagnostics
 end-of-job 9
 KDE 9
 KSE 7
 (*see also* error messages)
 format description errors
 (*see* format description diagnostics; error messages)
 format description number
 (*see* free-form)
 in header statement 5, 8
 selecting different format 11, 13, 55
 FORMAT NAME
 (*see* keyword-response summary charts)
 FORMAT NUMBER, SELECT
 (*see* keyword-response summary charts)
 FORMAT TYPE
 (*see also* keyword-response summary charts)
 formatted source statements 10
 FORMAT UNIT
 (*see* keyword-response summary charts)
 free-form
 KDE records 51
 KSE statements
 entering 24
 format of 11
 function key uses 2

 H 5, 8
 header statement 5, 8

 including another source module 12
 including statement numbers in KSE modules 4
 initialization keyword-response sequence 1
 KDE 48
 INPUT FILE TYPE 80
 inquiry interrupt 3
 insert a KSE statement 12

 KDE (Keyboard Data Entry) 47
 KDE accumulators 52
 KDE command key chart 54
 KDE control statements 59
 KDE data files 47
 KDE data records, entering 50
 KDE error messages 63

KDE format description 7, 20, 47
 KDE format description diagnostics 9
 KDE initialization 48
 KDE keyword-response summary chart 49
 KDE OCL cycle 62
 KDE sample jobs 66-75
 KDEFILE 47, 62
 key fields 47
 KDE supplies 47
 packed 47
 KEY FIELD LENGTH 47
 KEY FIELD START 47
 keyword 1
 keyword-response summary charts
 DIU 80
 KDE 49
 KSE
 creating format descriptions of KDE data records 20
 creating format descriptions of KSE source statements 18
 creating formatted source statements 22
 creating free-form source statements 24
 maintaining a procedure 25
 maintaining format descriptions of KDE data records 20
 maintaining format descriptions of KSE source statements 18
 maintaining formatted source statements 22
 maintaining free-form source statements 24
 KSE (Keyboard Source Entry) 1, 3
 KSE command key chart 11
 KSE command key restrictions 9, 11
 KSE control statements 26
 KSE error messages 28
 KSE format descriptions 5
 KSE format description diagnostics 9
 KSE formatted source statements 9
 KSE free-form source statements 11
 KSE keyword-response summary charts
 (see keyword-response summary charts, KSE)
 KSE source statements 9
 formatted 9
 free-form 11
 KSE source library module
 naming 5
 space available 5
 KSE statement numbering 4
 KSE work areas 4

label
 DIUFILE 83
 KDEFILE 47, 62

length
 field
 DIU 71
 KDE format description 7
 KSE format description 5
 key field
 (see key fields)
 line 7, 8, 9
 record, KDE 7, 47
 statement, KSE 5, 7
 line length 7, 8, 9
 line space key 3

LOAD
 DIU OCL response 83
 KDE OCL response 62
 KSE OCL response 32

maintaining
 (see keyword-response summary chart)
 messages, error
 DIU 84
 KDE 63
 KSE 28
 minus (—) key 2, 50
 module naming, KSE source 5
 module space available 5
 multiple KSE jobs 4

N (numeric field)
 DIU 77
 KDE 50
 KSE 10
 naming, source library module 5
 negative
 (see minus)
 NEW KDE FILE 48
 (see also KDE keyword-response summary chart)
 NEW SOURCE MODULE 5
 (see also keyword-response summary charts, KSE)
 numbering, KSE statement 4
 numeric fields
 DIU
 converting 79
 disk system management
 packed 77
 unpacked 77
 System/3 BASIC
 long precision 78
 short precision 78
 KDE 7, 50
 KSE 6, 10

OCL cycles
 DIU 83
 KDE 62
 KSE 32
 overflow, accumulator 52

P (packed fields)
 determining the line length 8
 DIU 77
 entering in KDE 50
 format description 8
 in key field 47
 packed numeric
 (see P)
 page width
 related to record length 8
 free-form records 51
 PASSWORD, BASIC 80

plus (+) key 2, 10, 50
previous field
 changing in KDE 53
previous KDE record
 correcting 53
 deleting 54
previous KSE statement
 correcting 11
 deleting 12
 duplicating from 9
procedures, control statements in
 (see control statements)
program start key 2
 indicate end-of-field 9, 50
 to enter free-form
 KDE 51
 KSE 11
 (see also keyword-response summary charts)
prompt 1

R
 (see unsigned numeric fields)
RECORD KEY, CORRECTING 54, 55
 (see also key fields)
RECORD KEY, DELETING 54, 57
 (see also key fields)
record length
 DIU 76
 KDE 7, 48
 KSE 5
 (see also diagnostics)
response 1
restarting KSE 4
restrictions, command key
 (see command key restrictions)
Return key 3
RPG II
 (see KSE sample jobs)
RPG II statement numbering 4
run type 4, 5

S (skip fields)
 entering KDE 50
 format descriptions of
 KDE 7
 KSE 5
sample jobs
 DIU 88
 KDE 66-75
 KSE 29-34, 33-46

SELECT FORMAT NUMBER
 (see FORMAT NUMBER, SELECT)
self-check fields 7, 50
SERIAL START POSITION 4
serializing KSE modules 4
signed numeric
 (see numeric fields)
significant digit 79
skip fields
 (see S)
skipping records 54
skipping forms 3
source library module 4
 naming 5
 space available 5
SOURCE MODULE NAME
 (see keyword-response summary charts)
SOURCE MODULE UNIT
 (see keyword-response summary charts)
source statements 9
 command key restrictions 11

 format descriptions of 5
START, KEY FIELD
 (see KEY FIELD START)
statement numbering, KSE 4
statement numbers
 KDE formats 8
 KSE formats 5, 7
STATEMENT NUMBER, CORRECTING 11, 13
STATEMENT NUMBER, DELETING 12, 14
System/3 BASIC
 converting 77-79
 data fields 63, 77-79

T
 (see self-check fields)
Tab Key 3

unpacked numeric fields
 DIU 77
unsigned numeric fields
 KDE 50
 KSE 10

work areas, KSE 4

YES, START 4
YES, SERIAL 4

zero-suppress numeric fields 10

00: response to SELECT FORMAT NUMBER
 (see free-form)

10000

10000

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

IBM System/3
Model 6
Conversational Utility Programs
Reference Manual

SC21-7528-4

YOUR COMMENTS, PLEASE . . .

Your comments assist us in improving the usefulness of our publications; they are an important part of the input used in preparing updates to the publications. All comments and suggestions become the property of IBM.

Please do not use this form for technical questions about the system or for requests for additional publications; this only delays the response. Instead, direct your inquiries or requests to your IBM representative or to the IBM branch office serving your locality.

Corrections or clarifications needed:

<i>Page</i>	<i>Comment</i>
-------------	----------------

Please include your name and address in the space below if you wish a reply.

● Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

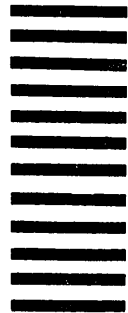
Cut Along Line

Fold

Fold

FIRST CLASS
PERMIT NO. 387
ROCHESTER, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES



POSTAGE WILL BE PAID BY . . .

IBM Corporation
General Systems Division
Development Laboratory
Rochester, Minnesota 55901

Attention: Publications, Dept. 245

Fold

Fold

IBM System/3 Model 6 Printed in U.S.A. SC21-7528-4

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 System/3 Model 6 Conversational Utility Programs
Program Number 5703-UT1 (System/3 Models 4 and 6)**

The IBM System/3 Model 6 Conversational Utility Programs used with System/3 Models 4 and 6 are a set of three disk-resident programs that converse with the operator by means of printed messages and questions.

1. Keyboard Source Entry Program

The Keyboard Source Entry (KSE) conversational utility program is used to create and maintain source library modules. Input to KSE is from the keyboard; output is to a source library module on disk. The input data is also printed to serve as a record of what was entered.

The KSE program has two run sequences: to create a source library module and to maintain an existing source library module.

Operations performed by KSE are controlled by the operator responses during the initial key-word-response sequence.

The following functions can be performed by KSE:

- Create a source library module of format descriptions of source statements.
- Maintain format descriptions of source statements in a source library module.
- Create a source library module of format descriptions of keyboard data entry records.
- Maintain format descriptions of keyboard data entry records in a source library module.
- Create a source library module of source statements.
- Maintain source statements in a source library module.
- Create and maintain a procedure in a source library module.

2. Keyboard Data Entry Program

The Keyboard Data Entry (KDE) conversational utility program processes data files for Models 4 and 6. The operator enters the data records using the keyboard. The KDE program processes the records, writes the data into an indexed data file, and prints the input data to record what was entered.

The KDE program has two run sequences: to create a disk data file (indexed) and to maintain an existing disk data file (indexed).

3. Data Interchange Utility Program (Model 6 only)

The Data Interchange Utility (DIU) program changes the format of System/3 BASIC data files so they can be used with the disk system management programs. Data files can also be converted to System/3 BASIC format. This program, therefore, allows programmers at installations with System/3 Model 6 co-resident programming system to convert data files created on one system to the format required to run on the other. The files are not converted in place, but instead a new file is created to contain the other data format.

This program is used only with the Model 6. (BASIC is not used with Model 4.)

Command Keys

The standard keyboard used with System/3 Models 4 and 6 contains a group of eight keys, called command keys, which are used during the operation of the conversational utility programs to control the functions being performed by the program.

Function Keys

Nine function keys on the keyboard are used when entering data into the system:

PROG START (program start)
ERASE (field erase)
BKSP (backspace)
Space bar
ENTER - (minus)
ENTER + (plus)
TAB (duplication) (KSE and KDE only)
RETURN (line space)
↑ (form skip) (appears as ! on display)

OCL Considerations

The KSE, KDE, and DIU conversational utility programs can be loaded using either the Operation Control Language (OCL) LOAD or CALL sequence. The CALL sequence requires the prior creation of a procedure using the OCL BUILD sequence.

Programming Systems

These programs operate under control of the System/3 Models 4 and 6 System Control Program (5703-SC1).

System Requirements

The minimum system configurations required by the Conversational Utility Programs are:

Model 4

- 5404 Processing Unit Model A18 (64K bytes) with Operator Keyboard Console
- 5447 Disk Storage and Control, one of the following:
 - Model A1
 - Model A2
- 5213 Printer Model 3
- 3277 Display Station Model 1 (for displaying CCP messages)
- At least one locally attached 3270 device

Model 6

- 5406 Processing Unit with Operator Keyboard Console, one of the following:
 - Model B3 (12K bytes)
 - Model B4 (16K bytes)
- 5444 Disk Storage Drive, one of the following:
 - Model 1
 - Model 2
 - One Model 2 and one Model 3
 - Two Model 2s
- Printer, one of the following:
 - 5213 Printer Model 1, 2, or 3
 - 2222 Printer Model 1 or 2

Reference Material

- *IBM System/3 Model 6 System/3 BASIC Reference Manual, GC34-0001*
- *IBM System/3 Models 4 and 6 Conversational Utility Programs Reference Manual, SC21-7528*

Compatibility

Data files created by the conversational utility programs can be accessed by any System/3 program that uses System/3 disk system data management. Data files that are converted by the Data Interchange Utility Program to System/3 BASIC format are not accessible by disk system management.

Programming Service Classification: C

The programming service classification assigned to any licensed program may be changed by IBM in accordance with the terms of the license agreement for IBM Program Products.

International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)