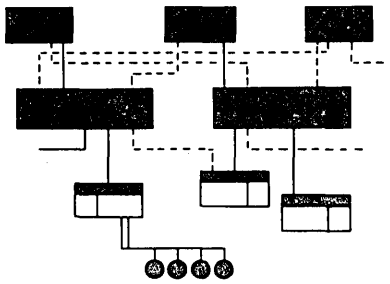
GENERAL ELECTRIC
COMPUTERS

GE-625/635 Integrated Data Store

SYSTEM
SUPPORT
INFORMATION



GENERAL ELECTRIC

CPB-1133A

GE-625/635
INTEGRATED
DATA STORE

February, 1966

Rev. September, 1966

GENERAL ELECTRIC

INFORMATION SYSTEMS DIVISION

PREFACE

The Integrated Data Store (IDS) system has been implemented as a set of over 35 subroutines, 12 of which represent the routines called directly by the user. The remaining subroutines are called internally and perform functions required within the system.

This manual describes all the subroutines constituting the IDS system.

Chapter 1 contains:

1. A description of the definition structure created by the IDS Translator from file characteristics contained in the user program.

2. A description of the calling sequence created by the IDS Translator from the user-supplied IDS verbs.

Chapter 2 is arranged alphabetically by subroutine name; the following items are given for each subroutine (when applicable):

1. Subroutine name.
2. A brief description of the function performed.
3. The form and content of the calling sequence.
4. The names of other subroutines which call the routine being described.
5. The names of the other subroutines which are called by the routine being described.
6. Restrictions on the use of the subroutine.
7. The size of the subroutine.
8. A list of errors detected by the subroutine.
9. A flow diagram indicating in general how the functions are performed.

The list below shows the order in which the IDS subroutines should be included in the system library:

.QDBUG	.QGET	.QMDFY	.QSDSW	.QUPDC	.QDLNK	.QBIC	.QCLOS
.QUTLA	.QGETC	.QMOVE	.QMNO	.QGDET	.QRUND	.QLOCK	.QUTLB
.QOPEN	.QGETD	.QSTOR	.QASC	.QAUTH	.QADJU	.QRLS	.QUTLC
.QCHN	.QGETE	.QUCCB	.QCALC	.QFWD	.QTYPX	.QPACK	
.QDLTE	.QHEAD	.QTLNK	.QSYN	.QUDCH	.QMRAC	.QUIT	

In this revised edition, changes in technical content from the previous edition are identified with a bar in the margin opposite the change.

Suggestions and criticisms relative to form, content, purpose, or use of this manual are invited. Comments may be sent on the Document Review sheet in back of this manual or may be addressed directly to Engineering Publications Standards, B-90, Computer Equipment Department, General Electric Company, 13430 North Black Canyon Highway, Phoenix, Arizona 85029.

CONTENTS

APPENDIXES

ILLUSTRATIONS

STRUCTURE DESCRIPTIONS

INTERPRETIVE SUBROUTINE INTERFACES

The IDS object program consists of a modular set of subroutines which interpretively execute the various IDS commands in response to the subroutine CALL's generated by the IDS Translator. To supplement the arguments of the various CALL's, the IDS routines depend upon a data description (hereafter referred to as the Definition Structure) to direct the action of the routines. This Definition Structure is also normally created by the IDS Translator.

The user who does not wish to use COBOL as his programming language can interface with IDS by coding his problem in the GE-625/635 Macro Assembler (GMAP) language including the IDS CALL's, the Definition Structure, and the IDS Communication Control Block, all of which are described in detail in this section. Each of the formats and calling sequences described in this section is the same as that generated by the IDS Translator.

DEFINITION STRUCTURE

The Definition Structure required by IDS is a list structure which reflects the description of the various data records of the IDS file. It defines the master/detail relationships between data records, the chain characteristics, and the physical and control characteristics of every field of every record type in the IDS file.

The organization of this Definition Structure is shown in Figure 1, using the IDS shorthand technique.

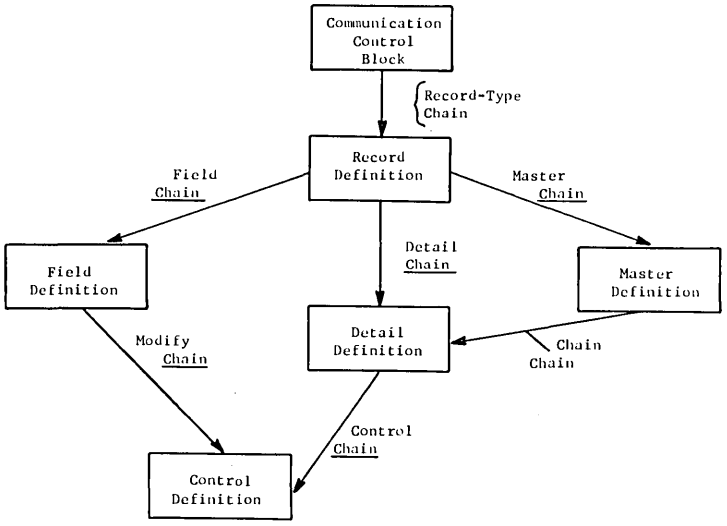
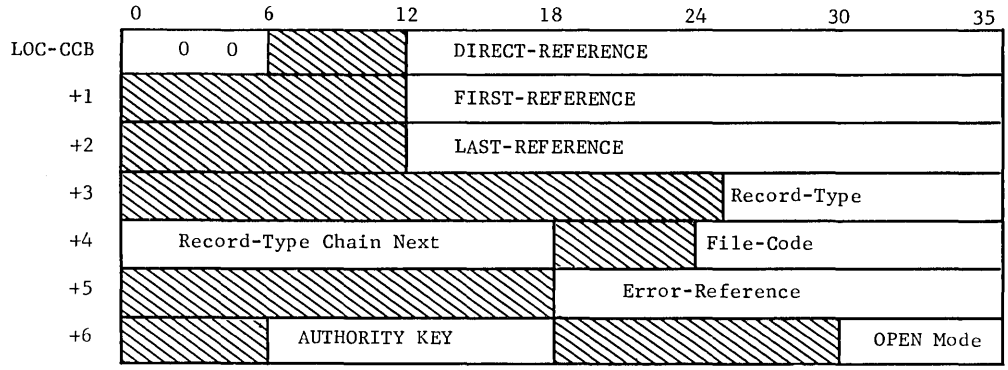


Figure 1. IDS Definition Structure

Each of the chains and entries of the above Definition Structure are described in detail in the following sections.

Communication Control Block

The single Communication Control Block entry must be supplied as the master of the Record-Type Chain. In addition, it serves as the communication area for certain data which must be passed between the user's program and the IDS subroutines. The machine format of the entry is shown in Figure 2.



(Note crosshatched areas are unused and must be zero.)

Figure 2. Machine Format of Communication Control Block Entry

The areas in the format shown in Figure 2 serve the following purposes:

LOC-CCB — Symbolic location of Communication Control Block.

0-5 — Definition type code--00 (octal)

12-35 — Reference code of the record last processed by any STORE or RETRIEVE routine. The value is supplied by an IDS routine or by the user's program, prior to the execution of the RETRIEVE DIRECT routine. (Binary integer)

LOC-CCB+1

12-35 — Reference code of the first or next record to be retrieved by the RETRIEVE EACH verb. The value is initially supplied by the user's program. After each retrieval, the IDS routine modifies the value to the reference code of the next record in the range. (Binary integer)

LOC-CCB+2

12-35 — Reference code of the last record of the range of records to be retrieved by the RETRIEVE EACH routine. The value is supplied by the user's program. (Binary integer)

LOC-CCB+3

26-35 — Record type of the last record processed by any STORE or RETRIEVE routine The value is supplied by an IDS routine. (Binary integer)

LOC-CCB+4

0-17 Record-Type Chain Next-Assigned symbol of the first Record Definition Structure.

24-35 The user-supplied two-character File Code assigned to the first mass storage device allocated to the IDS data file.

LOC-CCB+5

18-35 The BCD code for any error condition encountered during the execution of the last IDS routine. If there is no error, the value supplied by IDS will be zero.

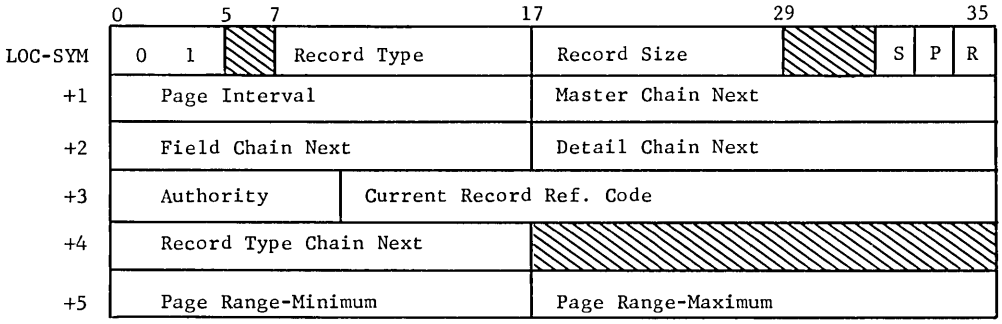
LOC-CCB+6

6-17 AUTHORITY KEY supplied via the OPEN routine.

30-35 Processing mode supplied via the OPEN routine.

Record Definition Entry

A Record Definition entry must be supplied for each data record type of the IDS data file. In addition, one such entry must be supplied for the Page Header record. The Record Definition entry is the master of the Master Chain, Detail Chain, and the Field Chain, and it is a detail of the Record-Type Chain. The machine format is shown in Figure 3.



(Note crosshatched areas are unused and must be zero.)

Figure 3. Machine Format of Record Definition Entry

The areas in the format shown in Figure 3 serve the following purposes:

LOC-SYM Symbol equivalent to the record name.

0-5 Definition type code--01 (octal)

8-17 Data Record Type--Must be within range of 1 to 999. Record type of Page Header record must be 1000. (Binary integer)

18-29 Data Record Size--Number of characters in the record including all control and chain fields. (See Appendix A.) (Binary integer)

33-S Storage Classification Indicator

0--Record is stored relative to the chain defined as the Retrieval Chain for this record. (See Detail Definition, page 5.)

1--Record is stored relative to the chain defined as the Storage Chain, which is not the same as the Retrieval Chain.

34-P Page Range Indicator

0--Page Range not specified for this record type.

1--Page Range is specified. (See LOC-SYM+5)

35-R Retrieval Classification

0--Secondary or calculated record

1--Primary record

LOC-SYM+1

0-17 Page Interval--Number of pages to be skipped relative to the page in which the last record of this type was stored. This only applies to primary or secondary records. (Binary integer)

18-35 Master Chain Next--Assigned symbol of the first Master Definition for this record. If this record is not the master of any chain, this is the assigned symbol of this Record Definition.

LOC-SYM+2

0-17 Field Chain Next--Assigned symbol of the first Field Definition for this record. If there are no data fields, then this is the assigned symbol of this Record Definition.

18-35 Detail Chain Next--Assigned symbol of the first Detail Definition for this record. If this record is not the detail in any chain, then this is the assigned symbol of this Record Definition.

LOC-SYM+3

0-11 Authority--A value supplied by the user not to exceed $4095_{(10)}$ which serves as a lock for the data contained in the record. Reference to this record during program execution is allowed only when a matching key is specified by the .QOPEN calling sequence. (Binary integer)

12-35 Current record reference code--Reference code of the last record stored or retrieved of this record type. This is supplied by IDS during execution.

LOC-SYM+4

0-17 Record Type Chain Next--Assigned symbol of the next Record Definition of the Definition Structure. If this is the last Record Definition entry, this field contains the symbolic location of the Communication Control Block.

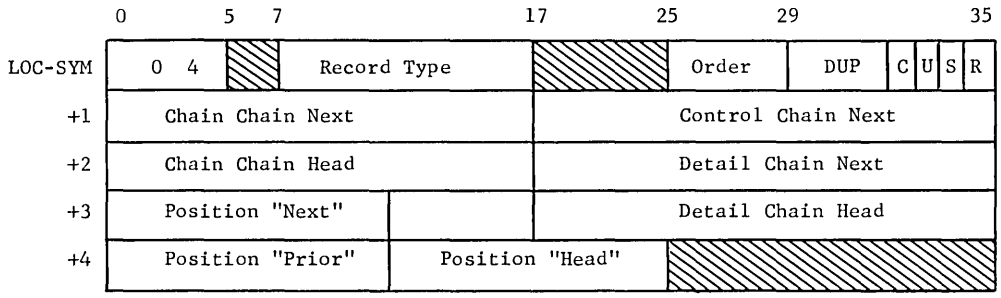
LOC-SYM+5 (This word required only if bit 34 of word zero equals - 1)

0-17 Page Range--Minimum--The first page number of a range of pages into which all records of this type are to be stored. (Binary integer)

18-35 Page Range--Maximum--The last page number of a range of pages into which all records of this type are to be stored. (Binary integer)

Detail Definition

A Detail Definition entry must be supplied each time a record is a detail in some chain. If a record is a detail in three different chains, three Detail Definition entries must be supplied. The Detail Definition entry is a detail of the Chain Chain and of the Detail Chain. It is also the master of the Control Chain. The machine format of this entry is shown in Figure 4.



(Note crosshatched areas are unused and must be zero.)

Figure 4. Machine Format of Detail Definition Entry

The areas in the format shown in Figure 4 serve the following purposes:

LOC-SYM Symbol assigned to this entry.

0-5 Definition Type--04 (octal)

8-17 Data Record Type--Same as that specified by the Record Definition entry for this record.

26-29 Order--A code to represent the chain order of the various details of this chain. Note that when several different record types are defined as details of the same chain, the chain-order must be the same for all records. The chain-order for a CALC chain must be $11_{(8)}$ for after current.

Octal Code	Chain-order
06	Sorted within Type
04	Sorted
10	First in Chain
00	Last in Chain
01	Before Current
11	After Current

30-31 DUP--Duplicate records indicator

00--Not allowed
01--Allowed First
11--Allowed Last

32 C--CALC Chain Detail Indicator

0--Not a CALC Chain
1--This is a CALC Chain

33 U--Chain Master Indicator

0--The master of this chain is a unique master retrievable via the MATCH-KEY fields defined for this chain.

1--The master of this chain is the current master record of its type.

34 S--Storage Chain Indicator

0--Record is not stored relative to this chain.
1--Record is stored relative to its logical position in this chain

35 R--Retrieval Chain Indicator

0--Associative retrieval of this record not possible via this chain.

1--Associative retrieval of this record must be via this chain.

LOC-SYM+1

0-17 Chain Chain Next--Assigned symbol of the next Detail Definition of this chain if there is more than one detail record type in the chain. If there is only one Detail Definition or if this is the last of several then this is the assigned symbol of the Master Definition for this chain.

18-35 Control Chain Next--Assigned symbol of the first Control Definition for this chain or, if none, the symbol assigned to this Detail Definition.

LOC-SYM+2

0-17 Chain Chain Head--Assigned symbol of the Master Definition of this chain.

18-35 Detail Chain Next--Assigned symbol of the next Detail Definition for this record if the record is a detail in more than one chain. If there is only one Detail Definition or if this is the last of several, then this is the assigned symbol of the Record Definition for this record.

LOC-SYM+3

0-11 Position "Next"--Character position, relative to the first character of the record, in which the first character of the chain next pointer is found. (Binary integer)

Note: If this is a CALC chain detail, the NEXT chain field must be the first field following the Record Size Field of the record; that is, it must be defined as beginning in character position 5.

18-35 Detail Chain Head--Assigned symbol of the Record Definition for this record.

LOC-SYM+4

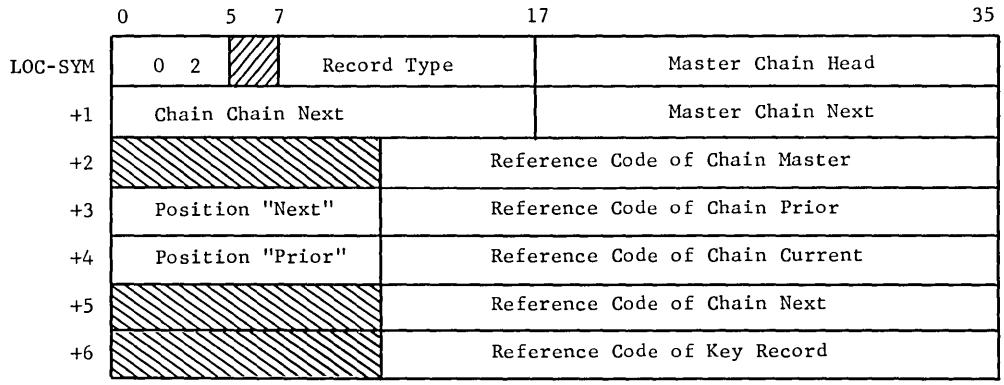
0-11 Position "Prior"--Character position, relative to the first character of the record, in which the first character of the chain prior pointer is found. If the chain is not prior processable, this value is zero. (Binary integer)

Note: When a detail record of a given chain contains a prior pointer all records of the chain must contain a prior pointer.

12-23 Position "Head"--Character position, relative to the first character of the record, in which the first character of the chain head pointer is found. If the chain is not a headed chain, this character is zero. (Binary integer)

Master Definition

A Master Definition entry must be supplied each time a record is defined as the master of some chain. The Master Definition is a detail of the Master Chain and the master of the Chain Chain. The machine format of this entry is shown in Figure 5.



(Note crosshatched areas are unused and must be zero.)

Figure 5. Machine Format for Master Definition Entry

The areas in the format shown in Figure 5 serve the following purposes:

LOC-SYM — Symbol equivalent to chain name.

0-5 — Definition Type--02 (octal)

8-17 — Data Record Type--Same as that specified for the Record Definition entry for this record.

18-35 — Master Chain Head--Assigned symbol of the Record Definition entry for this record.

LOC-SYM+1

0-17 — Chain Chain Next--Assigned symbol of the first Detail Definition for this chain. If the chain has no detail records defined, then this is the symbol of this Master Definition.

18-35 — Master Chain Next--Assigned symbol of the next Master Definition if this record is the master of more than one chain. If the record is the master of only one chain or the master of the last of several chains, then this coding is the symbol of the Record Definition for this record.

LOC-SYM+2

12-35 — Chain Table Master--Reference code of the Master Record of the chain defined by this Master Definition. This value is supplied by IDS during execution.

LOC-SYM+3

0-11 — Position "Next"--Character position, relative to the first character of the record, in which the first character of the chain next pointer is found. (Binary integer)

12-35 — Chain Table Prior--Reference code of the prior record of the chain defined by this Master Definition. This is supplied by IDS during execution.

LOC-SYM+4

0-11 — Position "Prior"--Character position, relative to the first character of the record, in which the first character of the chain prior pointer is found. If the master record is not prior processable, this value is zero. (Binary integer)

12-35 — Chain Table Current--Reference code of the current record of the chain defined by this Master Definition. This value is supplied by IDS during execution.

LOC-SYM+5

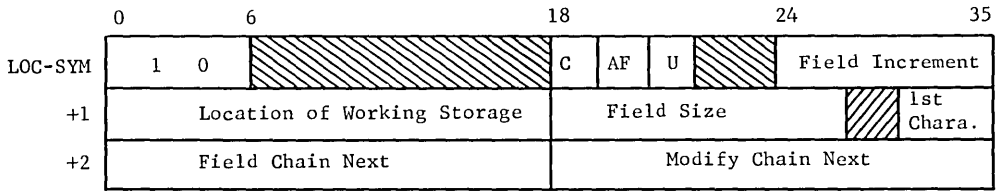
12-35 — Chain Table Next--Reference code of the next record of the chain defined by this Master Definition. This is supplied by IDS during execution.

LOC-SYM+6

12-35 Key Record--Reference code of the record to which a record will be relinked if there is an error in modification. This code is supplied by IDS during execution.

Field Definition

A Field Definition entry must be supplied for each data field contained in the record. (Note that Field Definitions are not supplied for the various chain fields; or for the reference code, record type, or record size fields). In addition, if the record is defined as a secondary record, a Field Definition must be supplied for all MATCH-KEY fields defined. If the record is defined as a primary record, a Field Definition must be supplied for the field which is equivalent to the reference code. The Field Definition entry is a detail in the Field Chain and is the master of the Modify Chain. The machine format of the entry is shown in Figure 6.



(Note the crosshatched areas are unused and must be zero.)

Figure 6. Machine Format for Field Definition Entry

The areas in the format shown in Figure 6 serve the following purposes:

LOC-SYM Symbol assigned to this entry.

0-5 Definition Type--10 (octal)

18 Computational Mode indicator (*)

0--Noncomputational field recorded in BCD.

1--Computational field recorded in Binary. (The implied size is 6 or 12 characters.)

Note: If data is stored in Binary form, compatibility of data formats with those of GE-400 Series Systems is not possible without special conversion routines in the GE-400 IDS.

19-20 AF--Arithmetic Form (*)--If Binary (bit 18=1)

00--Single Precision, Fixed Point
01--Single Precision, Floating Point
10--Double Precision, Fixed Point
11--Double Precision, Floating Point--If BCD (bit 18=0)

00--Alphanumeric
01--Alphabetic
10--Numeric
11--Signed numeric (sign indicated by zone bits of the low-order character of the field).

21 U--Unique Field Indicator

0--Field is not a unique or control field.

1--Field is unique and required for identification of the record.

Note: When this record is a primary record its unique field is, by definition, reference code. Since a Field Definition entry is not supplied for the reference code, a separate entry must be supplied to define the Working-Storage location for the field which is equivalent to reference code. This entry must not include the Field Definition specifications indicated in this section by (*), since the field is not actually contained in the data record. IDS assumes that the format of this field in Working-Storage is eight characters, BCD numeric, and synchronized left.

24-35 Field Increment (*)--Character position of the first character of a field, increment zero is the first character of the record. (Binary integer)

LOC-SYM+1

0-17 Location of Working-Storage--Assigned symbol of the leftmost word of Working-Storage defined for this field. The symbol is equivalent to the field name.

18-29 Field Size--The number of characters in the field as it exists in the record or in Working-Storage. (Binary integer)

33-35 First Character--Position of the first character of the field within the first word of Working-Storage. (Binary integer from 0 to 5)

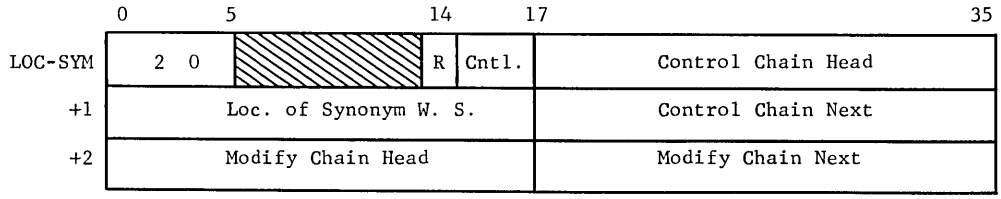
LOC-SYM+2

0-17 Field Chain Next--Assigned symbol of the next Field Definition of this record, if there is more than one field in the record. If there is only one field or if this is the last of several, then this value is the assigned symbol of the Record Definition for the record.

18-35 Modify Chain Next--Assigned symbol of the first Control Definition for this field or, if the field is not a control field, the symbol of this Field Definition.

Control Definition

A Control Definition entry must be supplied each time a field is defined as a control field of some chain. A control field is defined as a sort field, MATCH-KEY field, or a RANDOMIZE field. The Control Definition entry is a detail of the Modify Chain and of the Control Chain. The machine format of this entry is shown in Figure 7.



(Note the crosshatched areas are unused and must be zero.)

Figure 7. Machine Format of Control Definition Entry

The areas in the format shown in Figure 7 serve the following purposes:

LOC-SYM — Symbol assigned to this entry.

0-5 — Definition Type--20 (octal)

14 — R--Match Control indicator

0--Equality Match required
1--Match Equal or Greater (Range Record)

15-17 — CNTL--Control field type

001--RANDOMIZE control field
010--Sort Control--ascending sequence
011--Sort Control--descending sequence
100--MATCH-KEY control field

18-35 — Control Chain Head--Assigned symbol of the Detail Definition of the chain controlled by this Control Definition.

LOC-SYM+1

0-17 — Location of Synonym Working Storage--Assigned symbol of the first consecutive word of Working Storage defined for the synonym of this MATCH-KEY field. If there is no synonym this symbol is zero.

18-35 — Control Chain Next--Assigned symbol of the next Control Definition for the chain. If this is the last or only Control Definition, then the code is the symbol of the Detail Definition.

Note: When several sort control fields are defined for a given chain they must occur in sequence from major sort control to minor sort control.

LOC-SYM+2

0-17 — Modify Chain Head--Assigned symbol of the Field Definition for this control field.

18-35 — Modify Chain Next--Assigned symbol of the next Control Definition if this field is a control field in some other chain. If this is the last or only Control Definition for this field then the code is the symbol of the Field Definition.

CALLING SEQUENCES

The procedural STORE and RETRIEVE functions of IDS are executed by the use of a set of CALL's included in the user-prepared or translator-generated GMAP program. These CALL's closely parallel the various source language statements described in the explanation of the Procedure Division.

STORE Verb

The STORE Verb is used to place a given record into the IDS data file and to link the record to all chains with which the record is associated.

Call format:

```
CALL  .QSTOR (arg 1)
```

Where:

arg 1 is the symbolic location of the Record Definition of the record to be stored.

The .QSTOR routine assumes that the various Working-Storage fields defined for this record type have been initialized with the data values for each field of the record. The fields which uniquely defined the various chains involved are also assumed to have been initialized with their appropriate values. The record is stored as specified by the various parameters of the Record Definition.

Return is to the instruction following the CALL. At this point, the reference code of the new record is available in the Communication Control Block in word 0 as a 24-bit binary value. Also, the record type of the record is available in word 3 of the control block as a 10-bit binary value.

Within IDS, the reference code of this record has been recorded as the CURRENT record of its type and as the CURRENT record in each chain with which it is associated.

When a primary record is stored, the IDS-created reference code will have been stored in binary form in the DIRECT-REFERENCE field of the Communication Control Block. It will also have been stored in BCD form in the Working-Storage field which is equivalent to the reference code for the record.

Several error conditions are possible with the execution of this CALL. The user may determine the nature of the error, if any, by reference to word 5 of the Communication Control Block. If the content of this location is zero, no error exists. Otherwise, the low-order three characters of the word represent an error condition.

RETRIEVE Verb

The RETRIEVE Verb is used to cause a record to be made available in one of the IDS buffers. This action may or may not require that a page be transmitted from the mass storage device, since the page may already be in one of the buffers. No other action such as moving the record to Working-Storage is implied.

Five different calling sequences may be used to retrieve a record. These five provide for the seven record specifiers.

The first calling sequence is used to retrieve a record as specified by the "RETRIEVE data-name-1 RECORD" source language statement.

Call format:

```
CALL .QGET (arg-1)
```

Where:

arg-1 is the symbolic location of the Record Definition for this record type.

This routine assumes that all of the various control fields required to uniquely identify the record have been initialized in Working-Storage. If the record has been defined as a primary record, the data field which is equivalent to reference code must have been initialized.

If the record has been defined as a secondary record, all of the SORT and MATCH-KEY fields defined for the Retrieval Chain must have been initialized. If the record has been defined as a calculated record, all of the RAMDOMIZE control fields defined for the record must have been initialized.

The second retrieve calling sequence is used to retrieve a record as defined by the "RETRIEVE CURRENT" source language statement.

Call format:

```
CALL .QGETC (arg-1)
```

Where:

arg-1 is the symbolic location of the Record Definition for this record type.

The third retrieve calling sequence is used to retrieve a record as defined by the

"RETRIEVE { NEXT / PRIOR / MASTER } RECORD OF data-name-2 CHAIN"

source language statement.

Call format:

```
CALL .QCHN (arg-1, arg-2)
```

Where:

arg-1 is the symbolic location of the Master Definition for this chain.

arg-2 is a decimal integer indicating the option of the calling sequence.

1--NEXT of chain
2--PRIOR of chain
4--MASTER of chain

The fourth retrieve calling sequence is used to retrieve a record as specified by the "RETRIEVE EACH" source language statement.

Call format:

```
CALL .QGETE (arg-1)
```

Where:

arg-1 is the symbolic location of the procedure to which control will be transferred when the reference code of the record retrieved is greater than the reference code specified as the last of the range.

This routine assumes that prior to the first execution of the routine the user has initialized the Communication Control Block with the reference codes of the first and last record of the range to be searched. With each successful retrieval, the routine modifies the contents of the first-reference field in the control block to the next reference code in the range. When the routine finds that first reference is greater than the reference code of "last reference," the exit specified by argument-1 of the CALL is taken.

The last retrieve calling sequence is used to retrieve a record as specified by the "RETRIEVE DIRECT" source language statement.

Call format:

```
CALL .QGETD
```

In this case, the record identified by the reference code stored in word 0 of the Communication Control Block is retrieved.

With each of the above retrieve CALL's, return is to the instruction following the CALL. At this point, the reference code of the record is available in the Communication Control Block in word 0 as a 24-bit binary value. The reference code of the record retrieved has also been recorded as the CURRENT record of its type and as the CURRENT record in each chain with which it is associated.

Several error conditions are possible with the execution of this CALL. The user may determine the nature of the error, if any, by reference to word 5 of the Communication Control Block. If the content of this location is zero, no error exists. Otherwise, the low-order three characters of the word represent an error condition.

MOVE Statement

The MOVE statement is used to unpack data fields from the record last processed.

Call format:

```
CALL .QMOVE (arg-1, arg-2)
```

Where:

arg-1 is zero or the location of the first word of a Selective Field list. If arg-1 is zero, all fields of the record are moved to Working Storage.

arg-2 contains a binary integer in 0-17 equal to the number of fields to be moved. If arg-1 is zero, arg-2 is also zero.

Selective Field List format:

0-17 - Location of the Working Storage area for the field to be moved.
18-35 - Must be zero.

The data fields are unpacked according to the parameters defined for each Field Definition entry of the Field Chain for the record.

Normal return is to the instruction immediately following the CALL.

This routine checks the contents of ERROR-REFERENCE (word 5 of Communication Control Block) to determine whether any error occurred during or since the retrieval of the record involved. If this word is nonzero, the function of the routine is aborted and an immediate return to the instruction following the CALL is executed.

HEAD Statement

The HEAD statement is used to retrieve the master record of a given chain.

Call format:

```
CALL .QHEAD (arg-1)
```

Where:

arg-1 is the location of the Master Definition for the chain whose master record is to be retrieved.

Note that this routine also includes the automatic move of the fields of the record to Working-Storage.

This routine checks the contents of ERROR-REFERENCE (word 5 of Communication Control Block) to determine whether any error occurred during or since the retrieval of the record involved. If this word is nonzero, the function of the routine is aborted and an immediate return to the instruction following the CALL is executed.

Normal return is to the instruction following the CALL.

Several error conditions are possible with the execution of this CALL. The user may determine the nature of the error, if any, by reference to word 5 of the Communication Control Block. If the content of this location is zero, no error exists. Otherwise, the low-order three characters of the word represent an error condition.

MODIFY Statement

The MODIFY statement is used to modify by replacement the contents of specified fields of the record last processed and to maintain any chains which may be controlled by the fields modified.

Call format:

```
CALL .QMDFY (arg-1, arg-2)
```

Where:

arg-1 is the location of the first word of a list which defines the fields to be modified.

arg-2 contains a binary integer equal to the number of fields to be modified.

Field Modify List format:

0-17--Symbolic location of the Field Definition of the field to be modified.
18-35--Unused

This routine assumes that Working-Storage areas defined for the fields to be modified have been initialized with their respective change values prior to entry to this routine.

This routine checks the contents of ERROR-REFERENCE (word 5 of Communication Control Block) to determine whether any error occurred during or since the retrieval of the record involved. If this word is nonzero the function of the routine is aborted and an immediate return to the instruction following the CALL is executed.

Normal return is to the instruction immediately following the CALL.

Several error conditions are possible with the execution of this CALL. The user may determine the nature of the error, if any, by reference to word 5 of the Communication Control Block. If the content of this location is zero, no error exists. Otherwise, the low-order three characters of the word represent an error condition.

DELETE Statement

The DELETE statement is used to delete the record just previously retrieved with all of its dependent detail records. As an option, this statement may provide for interruption of the deletion process when a detail record of a given type is encountered in the data structure headed by the originally retrieved record.

Call format:

CALL .QDLTE (arg-1)

Where:

arg-1 contains the location of the first word of a Condition Action List, if this is a simple unconditional delete, the argument is zero.

Conditional Action List format:

First word of list.

0-17--Contains a binary integer equal to the number of additional entries in the list.
18-35--Must be zero.

The additional entries in the list are organized by record type in the sequence in which the various actions are to take place when the specified record type is encountered by the delete process.

The format of each of these entries is a function of the action to be accomplished.

"GO TO procedure-name-1"

0-17--Location of the transfer point.
18-23--01 (octal)
24-35--Record type expressed as a binary integer.

"MOVE TO WORKING STORAGE" (see MOVE Statement)

0-17--Location of the Record Definition of the record to be moved. This must be the record just encountered by the delete process.
18-23--02 (octal)
24-35--Record type expressed as a binary inte9er.

"HEAD data-name CHAIN" (see HEAD Statement)

0-17--Location of the Master Definition of the chain whose master record is to be returned. The chain must include the detail record just encountered by the delete process.
18-23--04 (octal)
24-35--Record type expressed as a binary integer.

"PERFORM procedure-name-2"

0-17--Must be zero.
18-23--10 (octal)
24-35--Record type expressed as a binary integer.

In the case of the PERFORM action, a sequence of four statements is required in the list and must immediately follow the word above. These statements are executed by the deletion process when the record type specified is encountered.

The statements are as follows:

```
ENTER COBOL
PERFORM procedure-name
ENTER GMAP

    TRA  *-2,I
```

Note that the subroutine executed returns to the ENTER GMAP statement, and the routine performed must not include any IDS statements.

The DELETE routine checks the contents of ERROR-REFERENCE (word 5 of Communication Control Block) to determine whether any error occurred during or since the retrieval of the record to be deleted. If this word is nonzero the function of the routine is aborted and an immediate return to the instruction following the CALL is executed.

Normal return is to the instruction following the CALL.

Several error conditions are possible with the execution of this CALL. The user may determine the nature of the error, if any, by reference to word 5 of the Communication Control Block. If the content of this location is zero, no error exists. Otherwise, the low-order three characters of the word represent an error condition.

OPEN Verb

The OPEN Verb must be used prior to use of any other IDS routine. This routine initializes the processing of the IDS data file.

Call format:

```
CALL .QOPEN (arg-1, arg-2, arg-3)
```

Where:

arg-1 is the location of the first word of the IDS Communication Control Block.

arg-2 is the location of a word of the following format:

```
ZERO A,M
```

Where: A is the AUTHORITY KEY to be used when accessing records from the IDS file.
M is a decimal value indicating the mode of processing.

1=Retrieve Only
2=Update

arg-3 is the location of the Page Header Record Definition entry.

This routine assumes that the Communication Control Block has been initialized with the file code of the first device assigned. If this parameter is not supplied, an error condition is noted in ERROR-REFERENCE (word 5 of the Communication Control Block) and subsequent references to the IDS file are rejected. Return is to the word following the CALL.

An attempt to open the IDS file when it is already open will result in no action and a direct return to the main program.

CLOSE Verb

The CLOSE Verb must be executed before the termination of the program which has referenced the IDS data file. This routine forces the writing of any page in memory which has been modified.

Call format:

```
CALL .QCLOS
```

Return is to the instruction following the CALL.

2. SUBROUTINE DESCRIPTIONS

Descriptions of the IDS subroutines with accompanying flow diagrams appear in alphabetical order in the rest of this chapter.

.QADJU

FUNCTION:

Subroutine .QADJU removes a given record from a page by moving all the following records in the page forward by the number of characters in the record removed. The page header record is altered as follows: the space available is increased, the available line number flag is reset, and the must-write switch is set.

CALLING SEQUENCE:

CALL .QADJU(ARG-1)

where ARG-1 is a word containing the reference code of the record to be removed.

CALLED BY:

.QFWD, .QSTOR, .QDLTE

CALLS FOR:

.QPACK

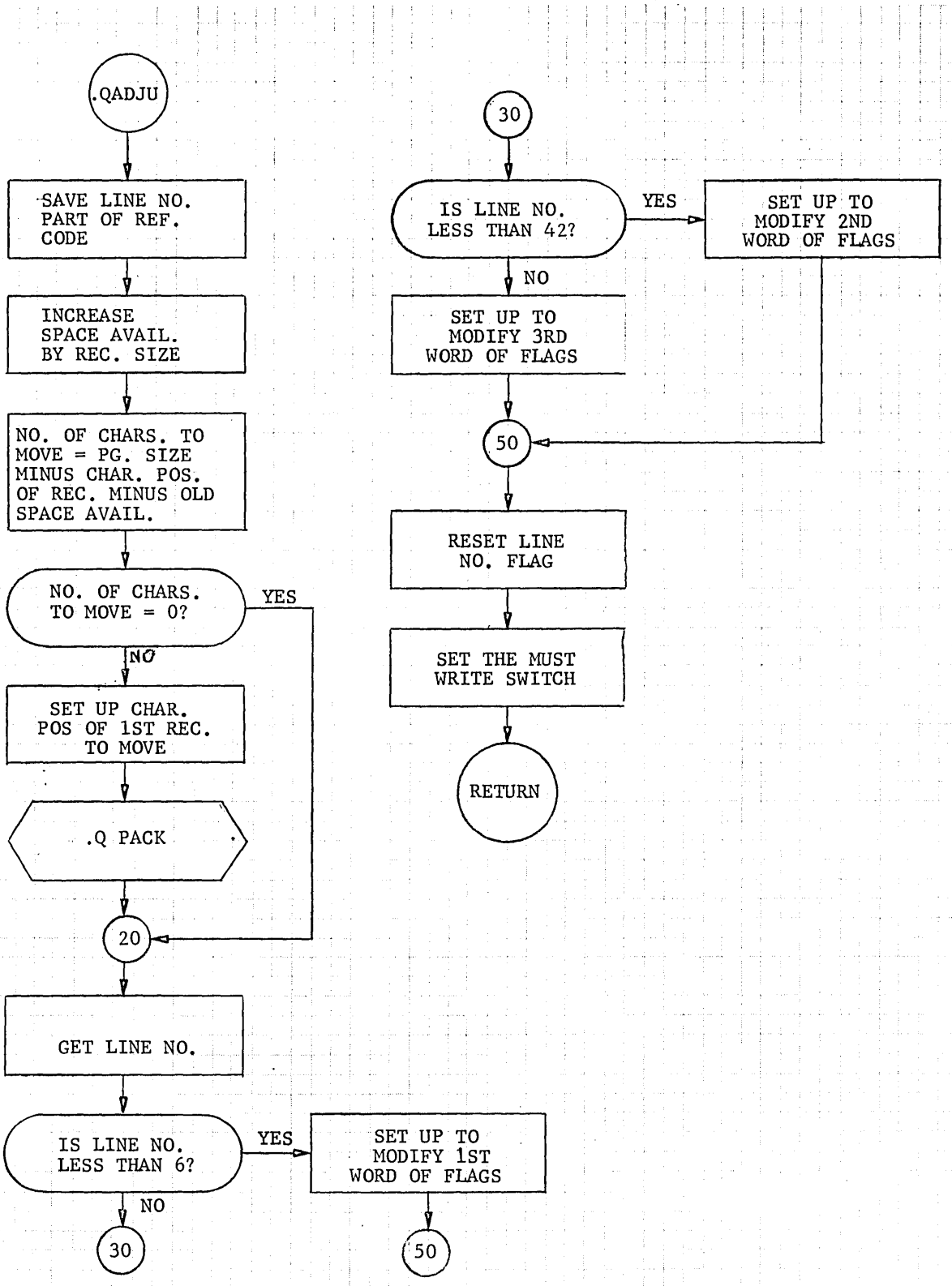
LIMITATIONS AND RESTRICTIONS:

Index register 2 must contain the location of the record definition of the record being removed.

The record to be removed must be the last one retrieved by .QMRAC.

NUMBER OF WORDS:

77 octal.


.QADJU
SAVE LINE NO. PART OF REF. CODE
INCREASE SPACE AVAIL. BY REC. SIZE
NO. OF CHARS. TO MOVE = PG. SIZE MINUS CHAR. POS. OF REC. MINUS OLD SPACE AVAIL.
NO. OF CHARS. TO MOVE = 0?
YES
NO
SET UP CHAR. POS OF 1ST REC. TO MOVE
.Q PACK
20
GET LINE NO.
IS LINE NO. LESS THAN 6?
YES
SET UP TO MODIFY 1ST WORD OF FLAGS
NO
30
50
30
IS LINE NO. LESS THAN 42?
YES
SET UP TO MODIFY 2ND WORD OF FLAGS
NO
SET UP TO MODIFY 3RD WORD OF FLAGS
50
RESET LINE NO. FLAG
SET THE MUST WRITE SWITCH
RETURN

.QASC

FUNCTION:

Subroutine .QASC retrieves a record according to the procedure defined by the retrieval clause of the data description. All control fields necessary for the unique identification of the record must be initialized in working storage.

CALLING SEQUENCE:

CALL .QASC

CALLED BY:

.QGET, .QTLNK

CALLS FOR:

.QCALC, .QSYN, .CNFXI, .QMRAC, .QUPDC, .QGDET, .QUIT

LIMITATIONS AND RESTRICTIONS:

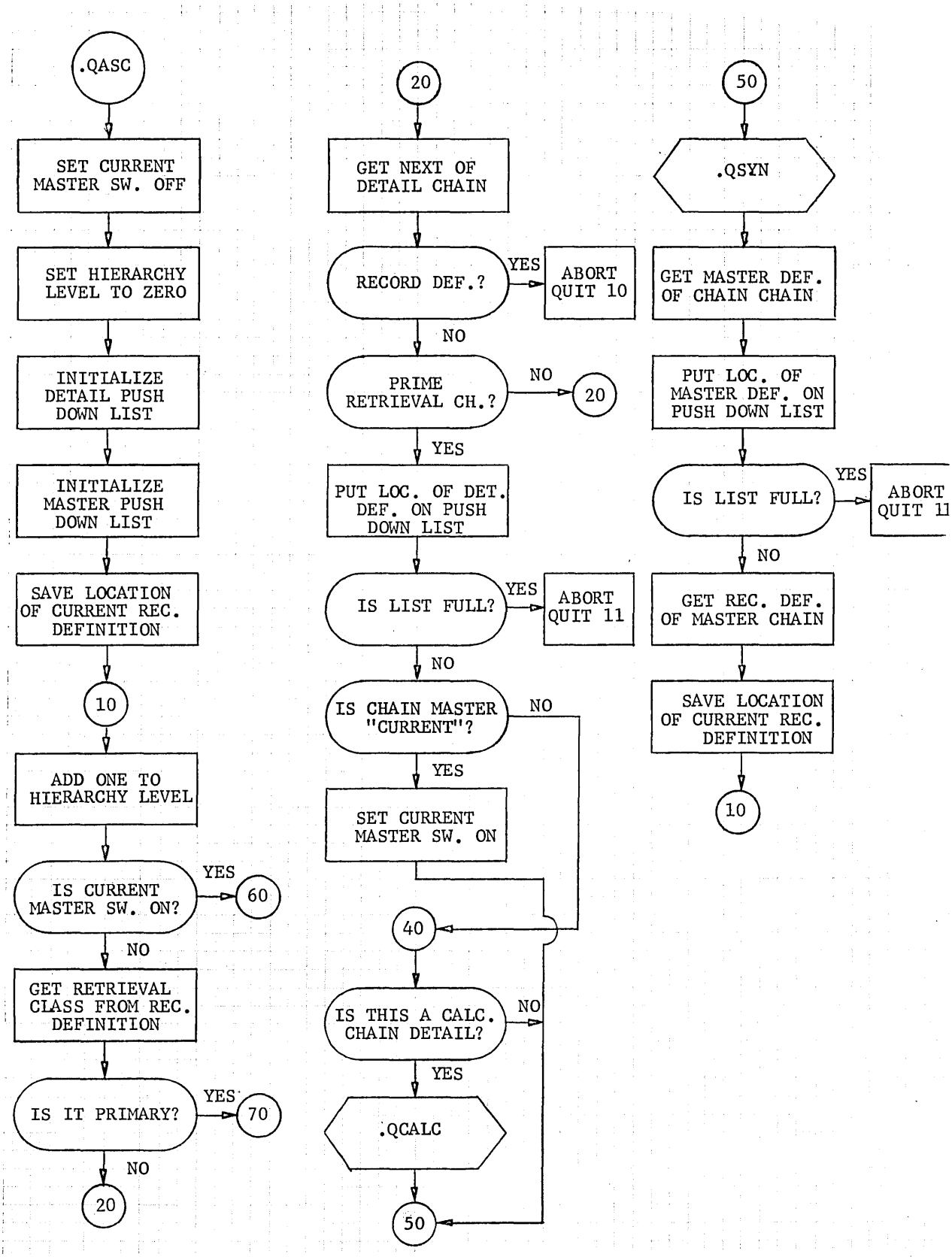
Index register 2 must contain the location of the record definition of the record being retrieved.

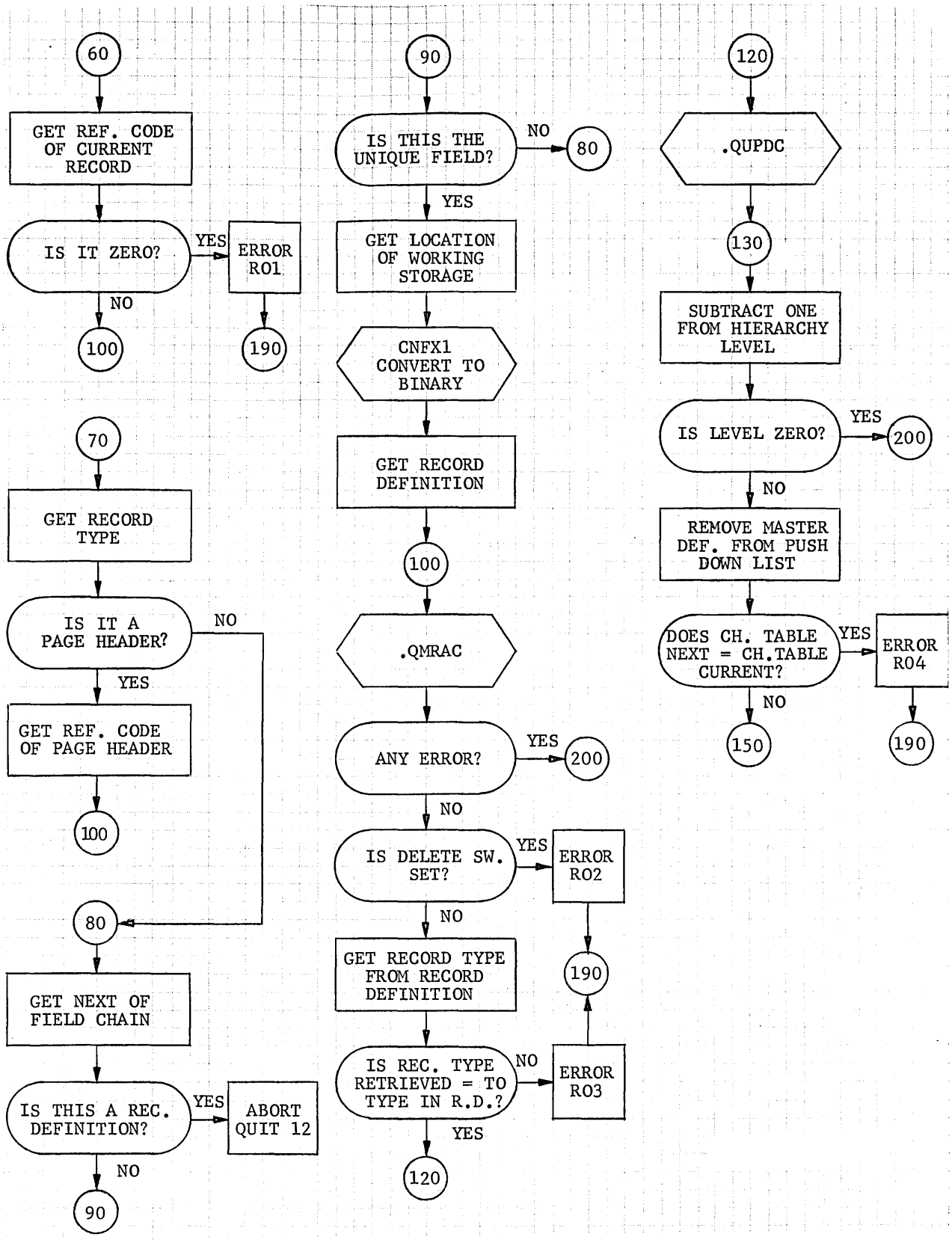
NUMBER OF WORDS:

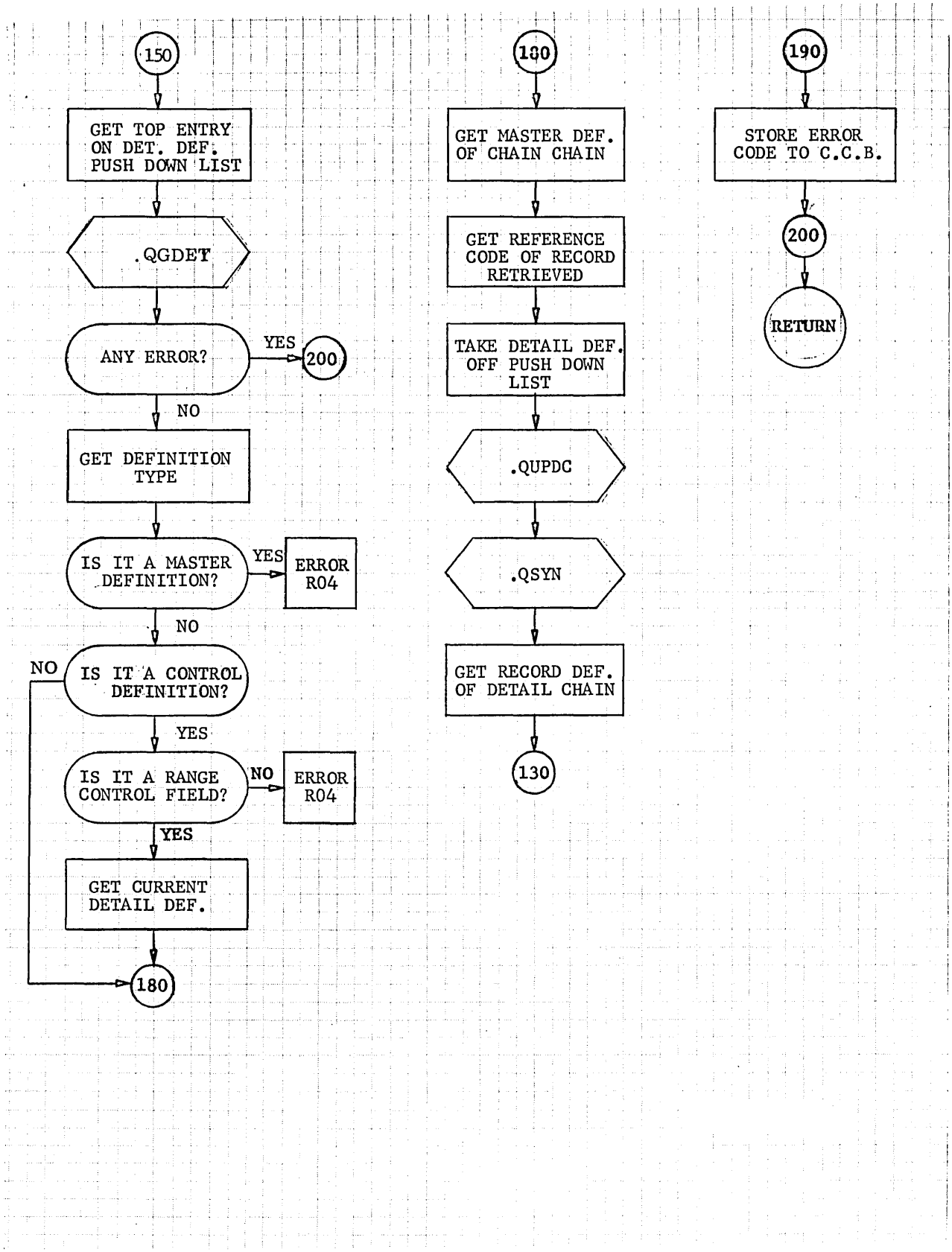
302 octal

DETECTED ERRORS:

QUIT (10)	No detail definition for secondary or calculated record.
QUIT (11)	IDS subroutine table overflow.
QUIT (12)	Unique field for primary record has not been defined.
R01	The retrieval of a record depends upon the selection of the current master of a given record type. That record has not been retrieved or has been deleted.
R02	A record cannot be retrieved, because the record or one of its masters has been deleted.
R03	The primary record retrieved is not the same record type as that specified in the record definition.
R04	An attempt was made to retrieve a record which does not exist within the data structure identified by supplied control fields.


.QASC
SET CURRENT MASTER SW. OFF
SET HIERARCHY LEVEL TO ZERO
INITIALIZE DETAIL PUSH DOWN LIST
INITIALIZE MASTER PUSH DOWN LIST
SAVE LOCATION OF CURRENT REC. DEFINITION
10
ADD ONE TO HIERARCHY LEVEL
IS CURRENT MASTER SW. ON?
YES
60
NO
GET RETRIEVAL CLASS FROM REC. DEFINITION
IS IT PRIMARY?
YES
70
NO
20
20
GET NEXT OF DETAIL CHAIN
RECORD DEF.?
YES
ABORT QUIT 10
NO
PRIME RETRIEVAL CH.?
NO
20
YES
PUT LOC. OF DET. DEF. ON PUSH DOWN LIST
IS LIST FULL?
YES
ABORT QUIT 11
NO
IS CHAIN MASTER "CURRENT"?
NO
YES
SET CURRENT MASTER SW. ON
40
IS THIS A CALC. CHAIN DETAIL?
NO
YES
.QCALC
50
50
.QSYN
GET MASTER DEF. OF CHAIN CHAIN
PUT LOC. OF MASTER DEF. ON PUSH DOWN LIST
IS LIST FULL?
YES
ABORT QUIT 11
NO
GET REC. DEF. OF MASTER CHAIN
SAVE LOCATION OF CURRENT REC. DEFINITION
10


60
GET REF. CODE OF CURRENT RECORD
IS IT ZERO?
YES
ERROR R01
NO
100
190
70
GET RECORD TYPE
IS IT A PAGE HEADER?
NO
YES
GET REF. CODE OF PAGE HEADER
100
80
GET NEXT OF FIELD CHAIN
IS THIS A REC. DEFINITION?
YES
ABORT QUIT 12
NO
90
90
IS THIS THE UNIQUE FIELD?
NO
80
YES
GET LOCATION OF WORKING STORAGE
CNFX1 CONVERT TO BINARY
GET RECORD DEFINITION
100
.QMRAC
ANY ERROR?
YES
200
NO
IS DELETE SW. SET?
YES
ERROR R02
NO
190
GET RECORD TYPE FROM RECORD DEFINITION
IS REC. TYPE RETRIEVED = TO TYPE IN R.D.?
NO
ERROR R03
YES
120
120
.QUPDC
130
SUBTRACT ONE FROM HIERARCHY LEVEL
IS LEVEL ZERO?
YES
200
NO
REMOVE MASTER DEF. FROM PUSH DOWN LIST
DOES CH. TABLE NEXT = CH.TABLE CURRENT?
YES
ERROR R04
NO
150
190


150
GET TOP ENTRY ON DET. DEF. PUSH DOWN LIST
.QGDET
ANY ERROR?
YES
200
NO
GET DEFINITION TYPE
IS IT A MASTER DEFINITION?
YES
ERROR R04
NO
NO
IS IT A CONTROL DEFINITION?
YES
IS IT A RANGE CONTROL FIELD?
NO
ERROR R04
YES
GET CURRENT DETAIL DEF.
180
130
GET MASTER DEF. OF CHAIN CHAIN
GET REFERENCE CODE OF RECORD RETRIEVED
TAKE DETAIL DEF. OFF PUSH DOWN LIST
.QUPDC
.QSYN
GET RECORD DEF. OF DETAIL CHAIN
130
190
STORE ERROR CODE TO C.C.B.
200
RETURN

.QAUTH

FUNCTION:

Subroutine .QAUTH checks the authority key supplied via the .QOPEN routine against the authority lock value defined for the record referenced by a Store, Modify, Delete, or Move routine.

Currently, the match is simply a test for equal values. If the values are not equal, the program will be aborted. IDS users will probably wish to apply unique matching conditions for their individual applications; this can be done by modifying this routine.

CALLING SEQUENCE:

```
CALL   .QAUTH
```

CALLED BY:

.QDLTE, .QMOVE, .QSTOR, .QMDFY

CALLS FOR:

.QUIT

LIMITATIONS AND RESTRICTIONS:

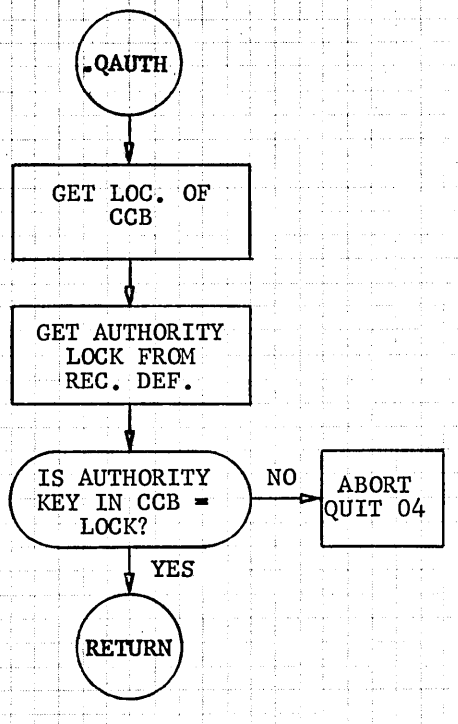
Index register 2 must contain the location of the record definition of the record referred to.

NUMBER OF WORDS:

22 octal.

DETECTED ERRORS:

QUIT (04)	Authority key does not match authority lock.


.QAUTH
GET LOC. OF CCB
GET AUTHORITY LOCK FROM REC. DEF.
IS AUTHORITY KEY IN CCB = LOCK?
NO
ABORT QUIT 04
YES
RETURN

.QBIC

FUNCTION:

Subroutine .QBIC retrieves a specified page and makes it available in core memory. .QCBUF contains the core location of the specified page. The pages already in core are searched first. If the page is not in core, it is read into the buffer specified by .QEMTY. Another buffer is then declared the .QEMTY buffer; and its page is written back to the disc if it has been modified since it was brought into core. The activity chain is updated.

CALLING SEQUENCE:

CALL .QBIC(ARG-1)

where ARG-1 is a word containing the 18-bit page number in 18-35.

CALLED BY:

.QMNO, .QMRAC

CALLS FOR:

.GREAD, .GWRIT, .GWAIT, .QLOCK, .QRLS

NUMBER OF WORDS:

212 octal.

ACTIVITY CHAIN LOGIC:

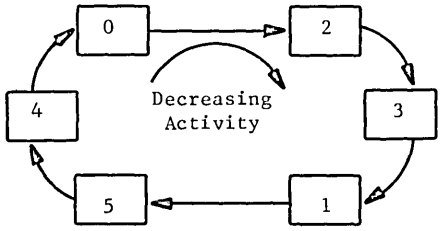
The activity chain is a closed circular loop of buffer numbers. The order of the chain is defined in an N word table, where N is the number of buffers available in core for storage of pages.

In the chart below, PRIOR is the buffer of next higher (more recent) activity; and NEXT is the buffer of next lower (less recent) activity.

At all times, one of the buffers is considered the MT buffer (the buffer of lowest activity). Its NEXT is considered the buffer of highest activity.

If 5 is the MT buffer in the drawing below, then 4 is the most active.

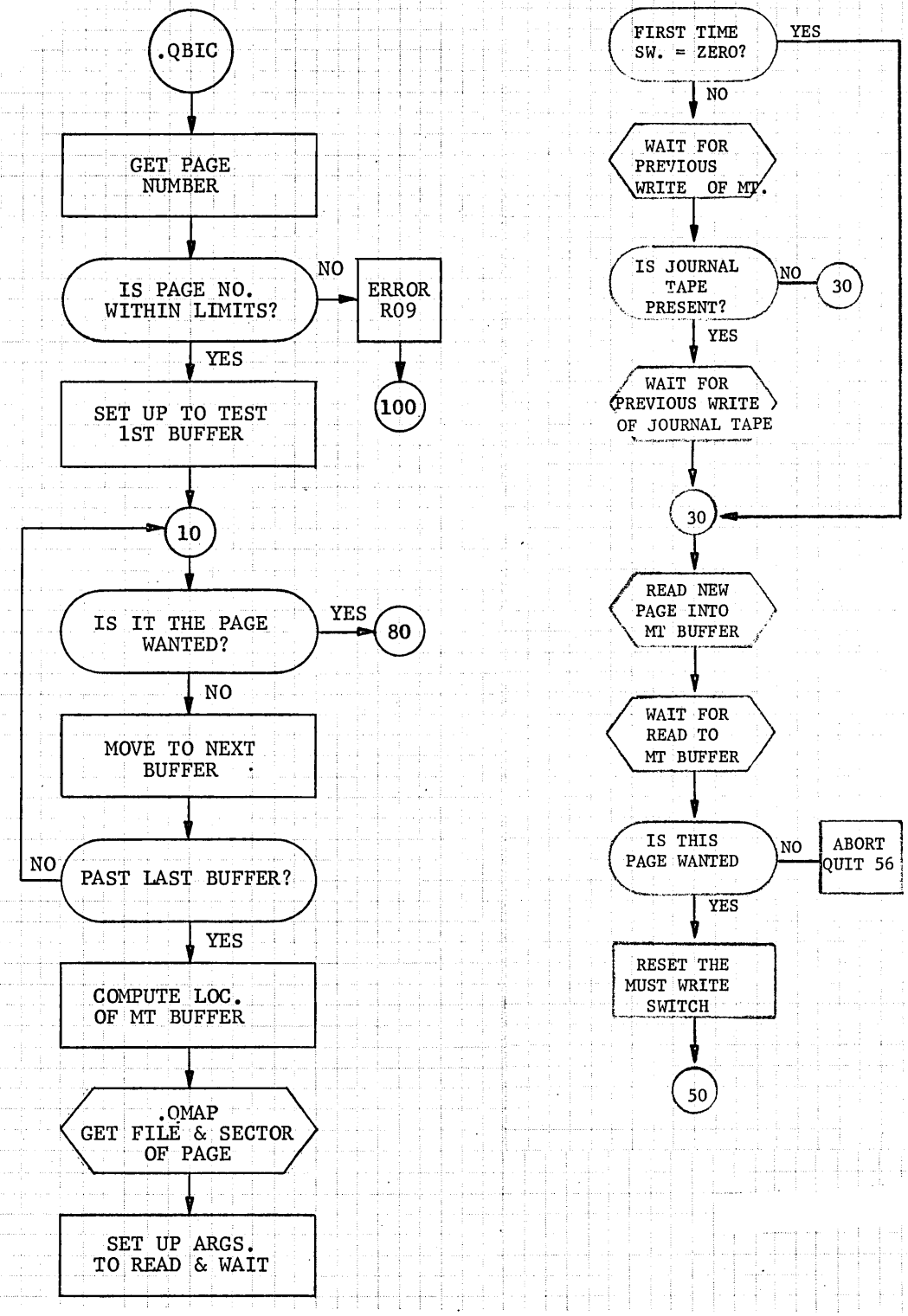
	PRIOR	NEXT
0	4	2
1	3	5
2	0	3
3	2	1
4	5	0
5	1	4

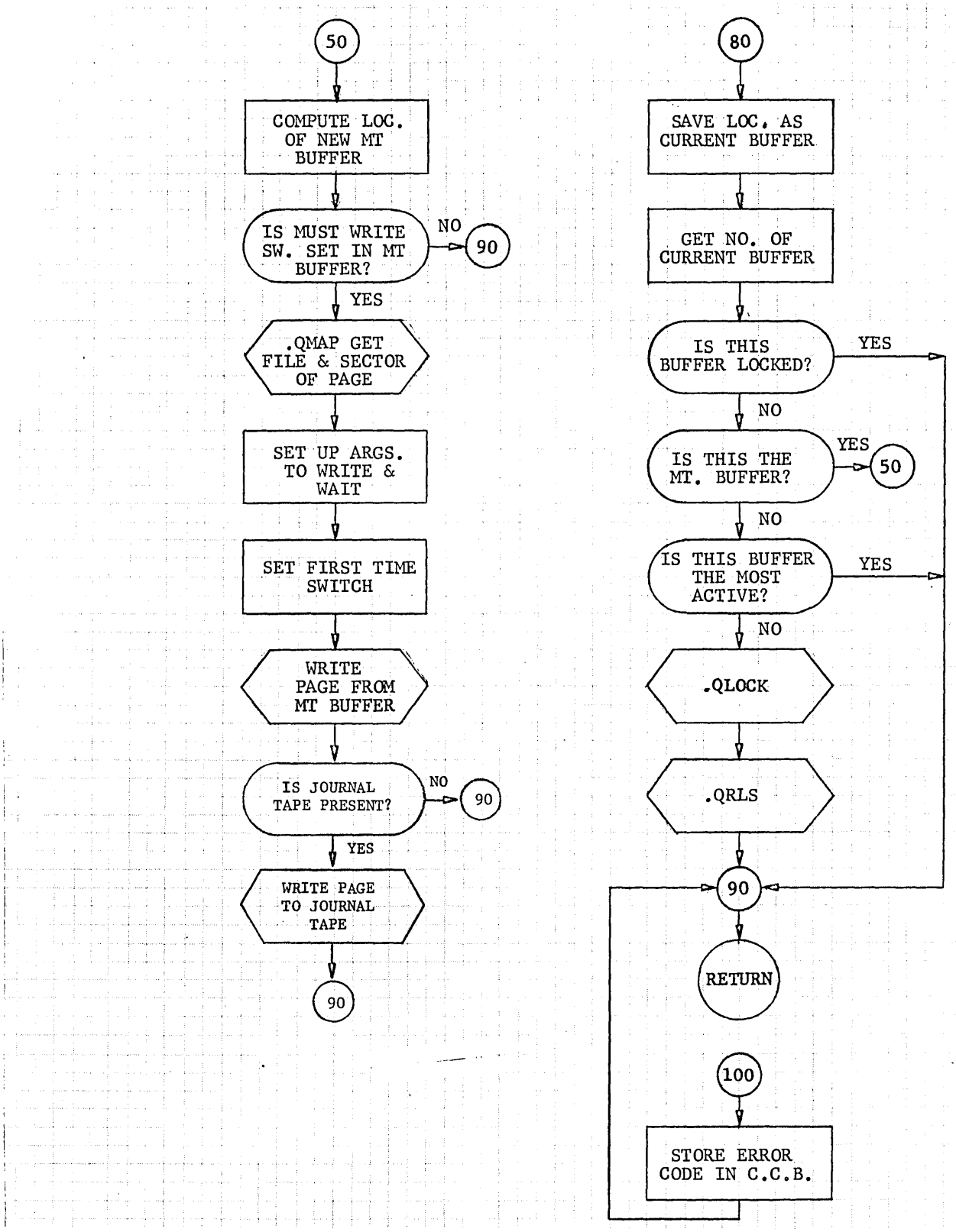


DETECTED ERRORS:

QUIT (56) Page read from mass storage device is not the page requested.

R09 Reference code of the record to be retrieved is outside of the total range of pages of the file.


.QBIC
GET PAGE NUMBER
IS PAGE NO. WITHIN LIMITS?
NO
ERROR R09
100
YES
SET UP TO TEST 1ST BUFFER
10
IS IT THE PAGE WANTED?
YES
80
NO
MOVE TO NEXT BUFFER
NO
PAST LAST BUFFER?
YES
COMPUTE LOC. OF MT BUFFER
.QMAP GET FILE & SECTOR OF PAGE
SET UP ARGS. TO READ & WAIT
FIRST TIME SW. = ZERO?
YES
NO
WAIT FOR PREVIOUS WRITE OF MT.
IS JOURNAL TAPE PRESENT?
NO
30
YES
WAIT FOR PREVIOUS WRITE OF JOURNAL TAPE
30
READ NEW PAGE INTO MT BUFFER
WAIT FOR READ TO MT BUFFER
IS THIS PAGE WANTED
NO
ABORT QUIT 56
YES
RESET THE MUST WRITE SWITCH
50


50
COMPUTE LOC. OF NEW MT BUFFER
IS MUST WRITE SW. SET IN MT BUFFER?
NO
90
YES
.QMAP GET FILE & SECTOR OF PAGE
SET UP ARGS. TO WRITE & WAIT
SET FIRST TIME SWITCH
WRITE PAGE FROM MT BUFFER
IS JOURNAL TAPE PRESENT?
NO
90
YES
WRITE PAGE TO JOURNAL TAPE
90
80
SAVE LOC. AS CURRENT BUFFER
GET NO. OF CURRENT BUFFER
IS THIS BUFFER LOCKED?
YES
NO
IS THIS THE MT. BUFFER?
YES
50
NO
IS THIS BUFFER THE MOST ACTIVE?
YES
NO
.QLOCK
.QRLS
90
RETURN
100
STORE ERROR CODE IN C.C.B.

.QCALC

FUNCTION:

Subroutine .QCALC produces a page header reference code by the repeated addition and shifting of the characters in the randomize control fields. The reference code is adjusted, so that it falls within the range of pages specified for the record type.

CALLING SEQUENCE:

```
CALL   .QCALC
```

CALLED BY:

.QASC, .QTLNK

CALLS FOR:

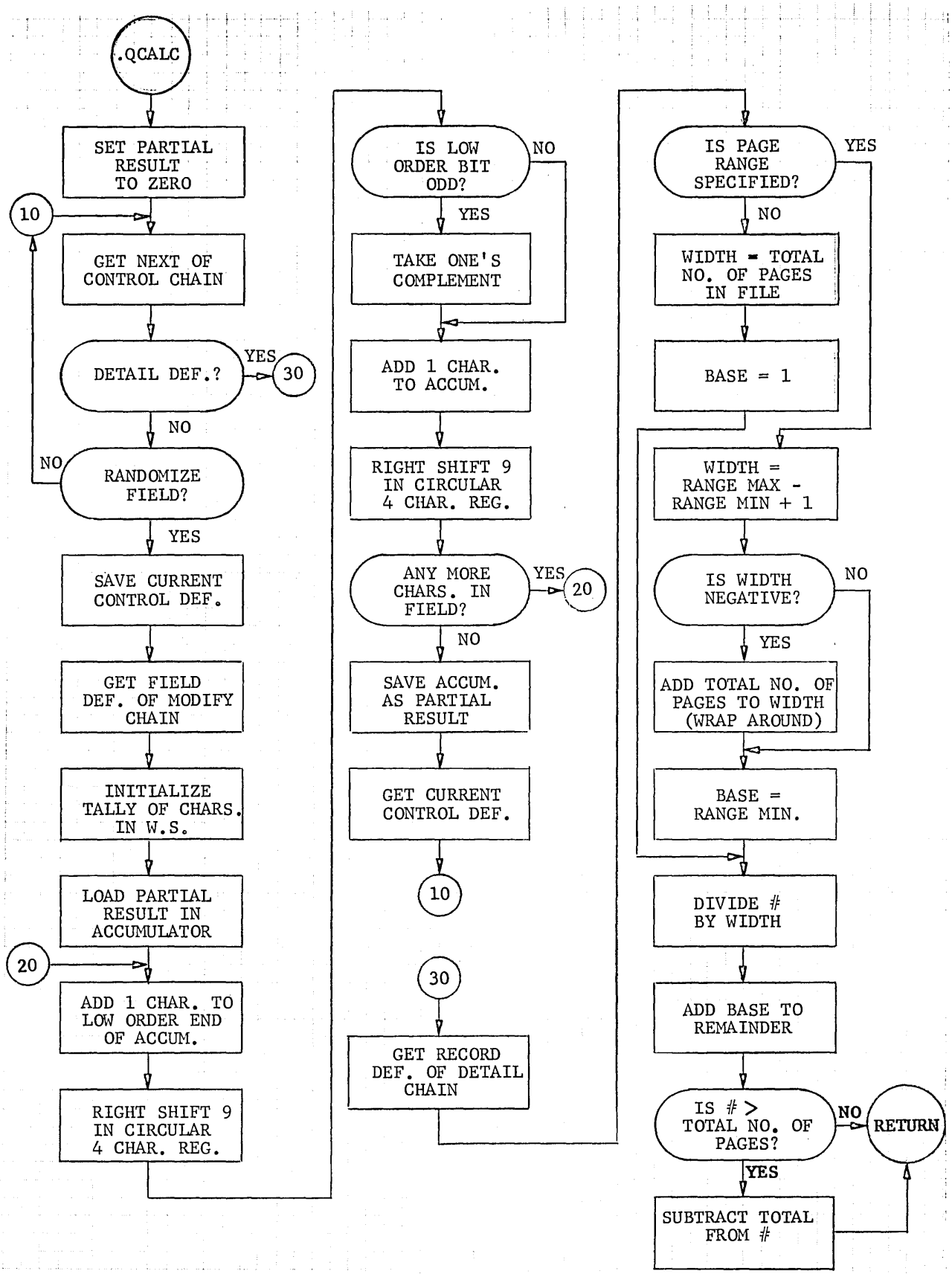
Not applicable

LIMITATIONS AND RESTRICTIONS:

Index register 2 must contain the location of the detail definition that designates the record as a detail of the CALC chain.

NUMBER OF WORDS:

117 octal.


.QCALC
SET PARTIAL RESULT TO ZERO
10
GET NEXT OF CONTROL CHAIN
DETAIL DEF.?
YES
30
NO
NO
RANDOMIZE FIELD?
YES
SAVE CURRENT CONTROL DEF.
GET FIELD DEF. OF MODIFY CHAIN
INITIALIZE TALLY OF CHARS. IN W.S.
LOAD PARTIAL RESULT IN ACCUMULATOR
20
ADD 1 CHAR. TO LOW ORDER END OF ACCUM.
RIGHT SHIFT 9 IN CIRCULAR 4 CHAR. REG.
IS LOW ORDER BIT ODD?
NO
YES
TAKE ONE'S COMPLEMENT
ADD 1 CHAR. TO ACCUM.
RIGHT SHIFT 9 IN CIRCULAR 4 CHAR. REG.
ANY MORE CHARS. IN FIELD?
YES
20
NO
SAVE ACCUM. AS PARTIAL RESULT
GET CURRENT CONTROL DEF.
10
30
GET RECORD DEF. OF DETAIL CHAIN
IS PAGE RANGE SPECIFIED?
YES
NO
WIDTH = TOTAL NO. OF PAGES IN FILE
BASE = 1
WIDTH = RANGE MAX - RANGE MIN + 1
IS WIDTH NEGATIVE?
NO
YES
ADD TOTAL NO. OF PAGES TO WIDTH (WRAP AROUND)
BASE = RANGE MIN.
DIVIDE # BY WIDTH
ADD BASE TO REMAINDER
IS # > TOTAL NO. OF PAGES?
NO
RETURN
YES
SUBTRACT TOTAL FROM #

.QCHN

FUNCTION:

Subroutine .QCHN retrieves a record which is next or prior or master, relative to the current record of the specified chain. The current record of the chain is defined by the chain table contained in the master definition. If the chain was not previously referred to, an error condition is noted when control is returned.

CALLING SEQUENCE:

CALL .QCHN(ARG-1, ARG-2)

where ARG-1 is the location of the master definition of the chain.

ARG-2 is a decimal value indicating the function of the call:

1 = Next
2 = Prior
3 = Master

CALLED BY:

Source program

CALLS FOR:

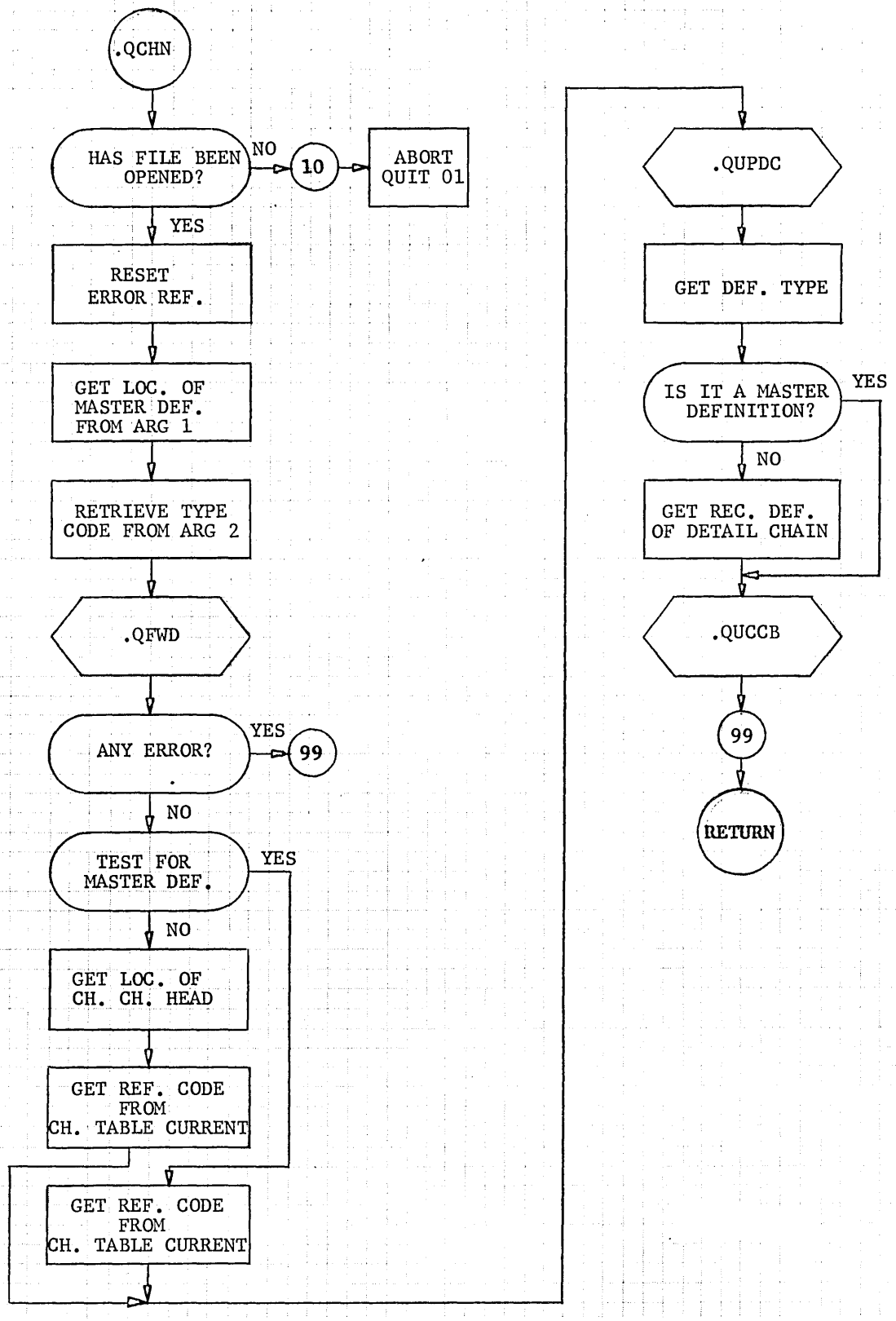
.QFWD, .QUPDC, .QUCCB, .QUIT

NUMBER OF WORDS:

62 octal

DETECTED ERRORS:

QUIT (01) Attempt to store or retrieve a record with the IDS file not opened.


.QCHN
HAS FILE BEEN OPENED?
NO
10
ABORT QUIT 01
YES
RESET ERROR REF.
GET LOC. OF MASTER DEF. FROM ARG 1
RETRIEVE TYPE CODE FROM ARG 2
.QFWD
ANY ERROR?
YES
99
NO
TEST FOR MASTER DEF.
YES
NO
GET LOC. OF CH. CH. HEAD
GET REF. CODE FROM CH. TABLE CURRENT
GET REF. CODE FROM CH. TABLE CURRENT
.QUPDC
GET DEF. TYPE
IS IT A MASTER DEFINITION?
YES
NO
GET REC. DEF. OF DETAIL CHAIN
.QUCCB
99
RETURN

.QCLOS

FUNCTION:

Subroutine .QCLOS writes back to the disc any of the pages in memory which have been modified since they were brought into core.

CALLING SEQUENCE:

CALL .QCLOS

CALLED BY:

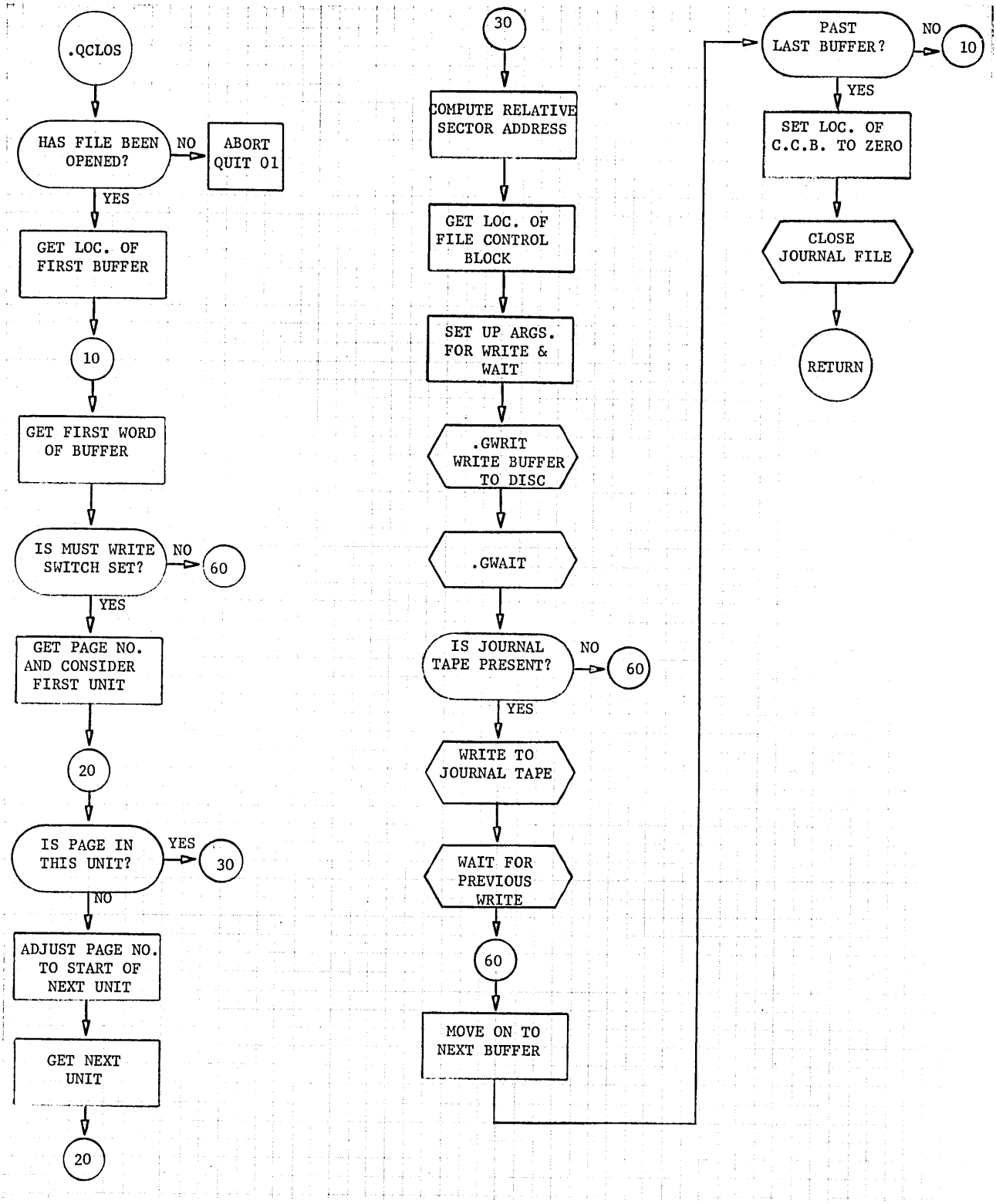
Source program

CALLS FOR:

.GWRIT, .GWAIT

NUMBER OF WORDS:

61 octal.


.QCLOS
HAS FILE BEEN OPENED?
NO
ABORT QUIT 01
YES
GET LOC. OF FIRST BUFFER
10
GET FIRST WORD OF BUFFER
IS MUST WRITE SWITCH SET?
NO
60
YES
GET PAGE NO. AND CONSIDER FIRST UNIT
20
IS PAGE IN THIS UNIT?
YES
30
NO
ADJUST PAGE NO. TO START OF NEXT UNIT
GET NEXT UNIT
20
30
COMPUTE RELATIVE SECTOR ADDRESS
GET LOC. OF FILE CONTROL BLOCK
SET UP ARGS. FOR WRITE & WAIT
.GWRIT WRITE BUFFER TO DISC
.GWAIT
IS JOURNAL TAPE PRESENT?
NO
60
YES
WRITE TO JOURNAL TAPE
WAIT FOR PREVIOUS WRITE
60
MOVE ON TO NEXT BUFFER
PAST LAST BUFFER?
NO
10
YES
SET LOC. OF C.C.B. TO ZERO
CLOSE JOURNAL FILE
RETURN

.QDBUG

FUNCTION:

Subroutine .QDBUG allows the user to dump specified portions of core during the processing of an IDS application.

CALLING SEQUENCE:

```
CALL   .QDBUG (arg-1, arg-2)
```

Where arg-1 is a word containing the location of the first word of a conditional action list (see below); arg-2 is the number of entries contained in the list.

CALLED BY:

Source program

CALLS FOR:

.GOPEN, .GPRNT, EPRINT, .GCLSE, .QCHN

NUMBER OF WORDS:

1673 octal

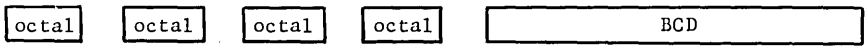
CONDITIONAL ACTION LIST FORMAT:

	Parameter	Code	
LIST		01	Current buffer
+1		02	Current record
+2		03	Communication control block
+3	MD00XX	04	Chain tables for chain X
+4	MD00XX	05	Records contained in chain X

EXECUTION OUTPUT:

The printouts produced by the DEBUG Verb will be written on the system output file for printing on the execution report. Output is in the following formats:

PAGE XXX Calling Address



Side-by-side octal/BCD printout of the current page:

RECORD Ref. Code Record Type Calling Address

Side-by-side octal/BCD printout of the current record:

COMMUNICATION CONTROL BLOCK Calling Address

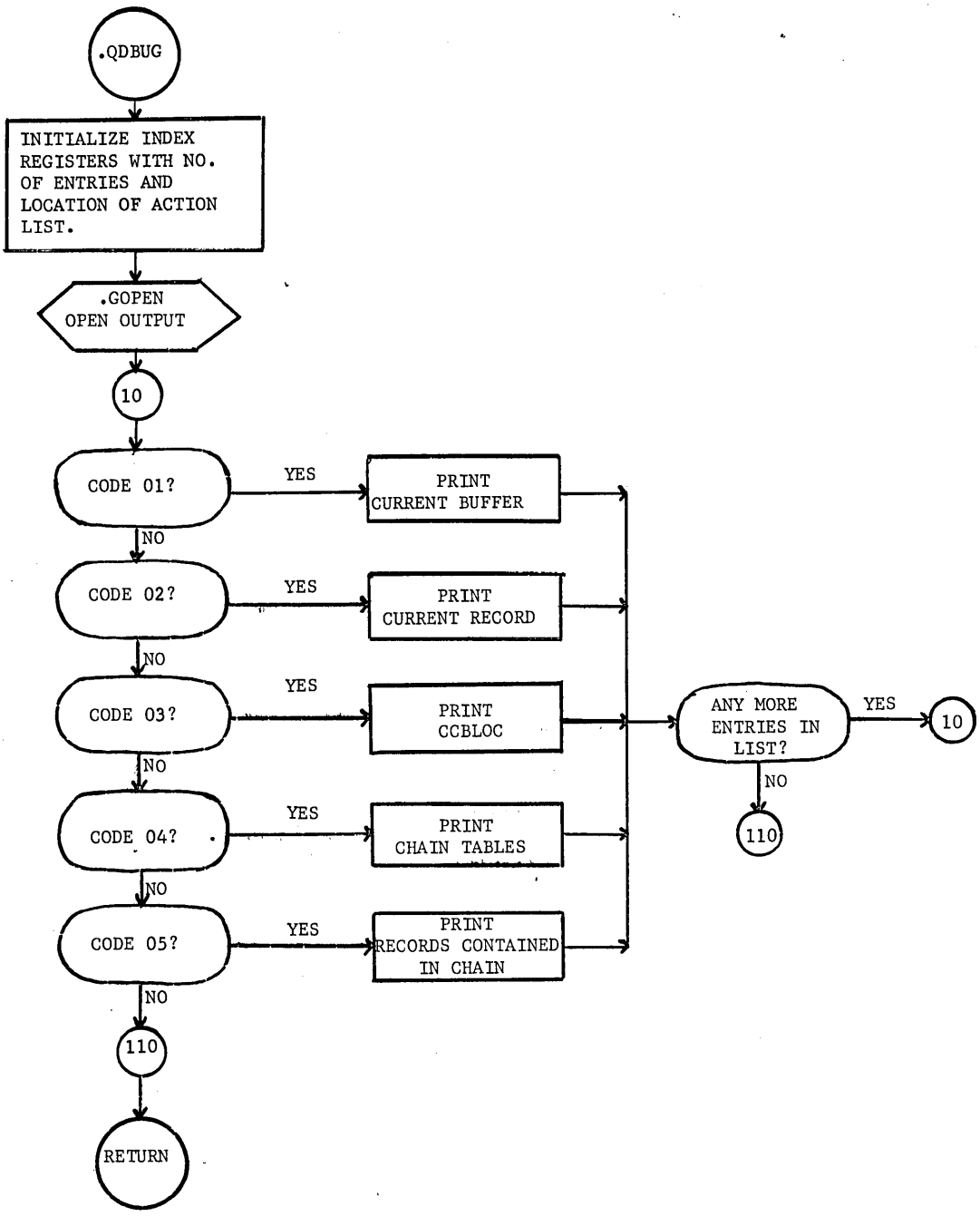
DIRECT REFERENCE	Ref. Code in octal	Ref. Code in BCD
FIRST REFERENCE	Ref. Code in octal	Ref. Code in BCD
LAST REFERENCE	Ref. Code in octal	Ref. Code in BCD
RECORD TYPE	Rec. Type in octal	Rec. Type in BCD
ERROR REFERENCE	Error Code in BCD	

Chain-name-2 CHAIN Calling Address

CHAIN TABLE HEAD	Ref. Code in octal	Ref. Code in BCD
CHAIN TABLE PRIOR	Ref. Code in octal	Ref. Code in BCD
CHAIN TABLE CURRENT	Ref. Code in octal	Ref. Code in BCD
CHAIN TABLE NEXT	Ref. Code in octal	Ref. Code in BCD

Chain-name-3 CHAIN Calling Address

Side-by-side octal/BCD printout of the records contained within the specified chain.


.QDBUG
INITIALIZE INDEX REGISTERS WITH NO. OF ENTRIES AND LOCATION OF ACTION LIST.
.GOPEN
OPEN OUTPUT
10
CODE 01?
YES
PRINT CURRENT BUFFER
NO
CODE 02?
YES
PRINT CURRENT RECORD
NO
CODE 03?
YES
PRINT CCBLOC
NO
CODE 04?
YES
PRINT CHAIN TABLES
NO
CODE 05?
YES
PRINT RECORDS CONTAINED IN CHAIN
NO
110
RETURN
ANY MORE ENTRIES IN LIST?
YES
10
NO
110

.QDLNK

FUNCTION:

Subroutine .QDLNK delinks from the specified chain the detail record whose reference code is contained in the chain table CURRENT. The chain table is then updated, so that the new chain table CURRENT is the record prior to the one delinked.

CALLING SEQUENCE:

```
CALL   .QDLNK
```

CALLED BY:

.QWFD, .QTLNK

CALLS FOR:

.QRUND, .QTYPX, .QPACK, .QUIT

LIMITATIONS AND RESTRICTIONS:

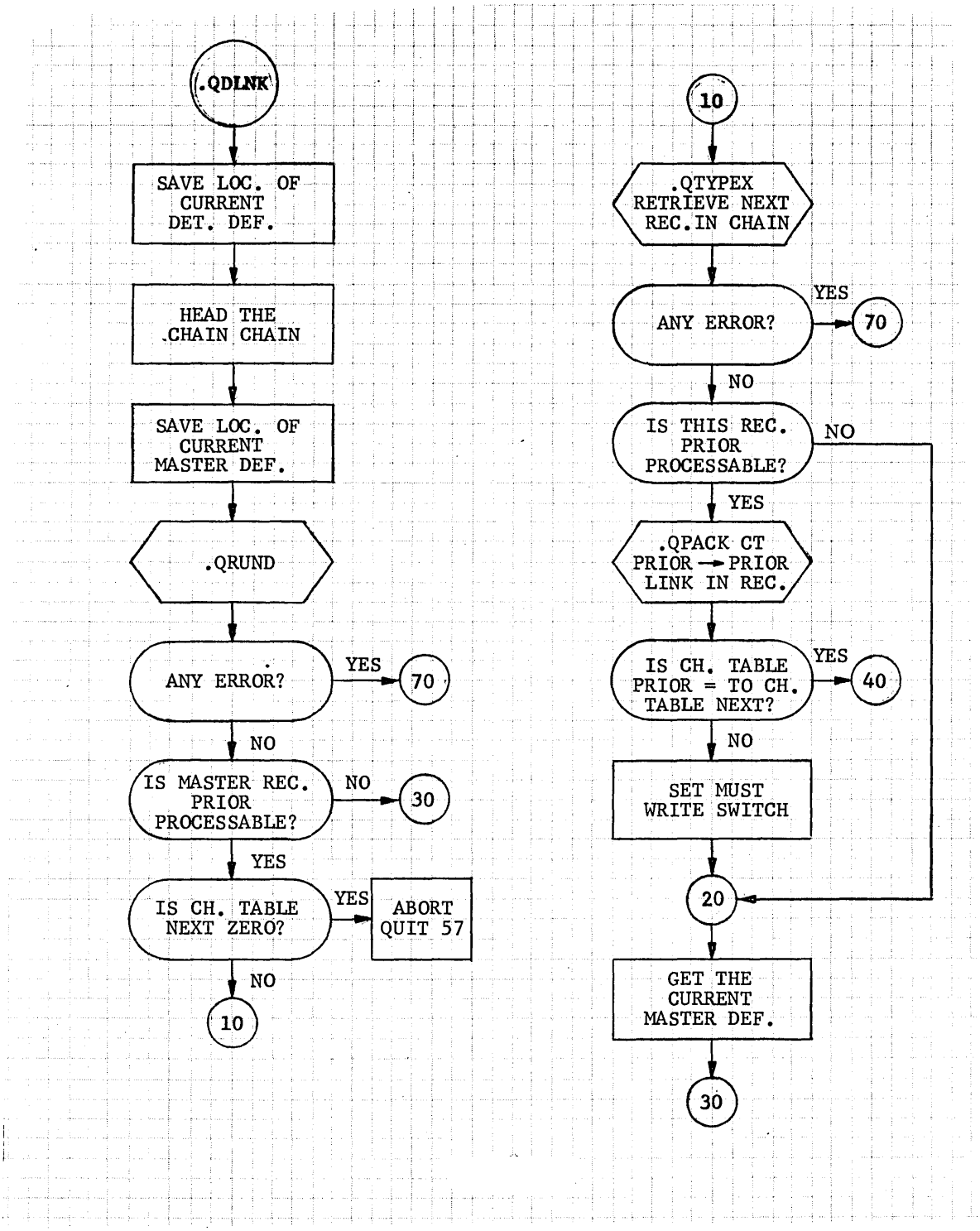
Index register 2 must contain the location of the detail definition specifying the chain.

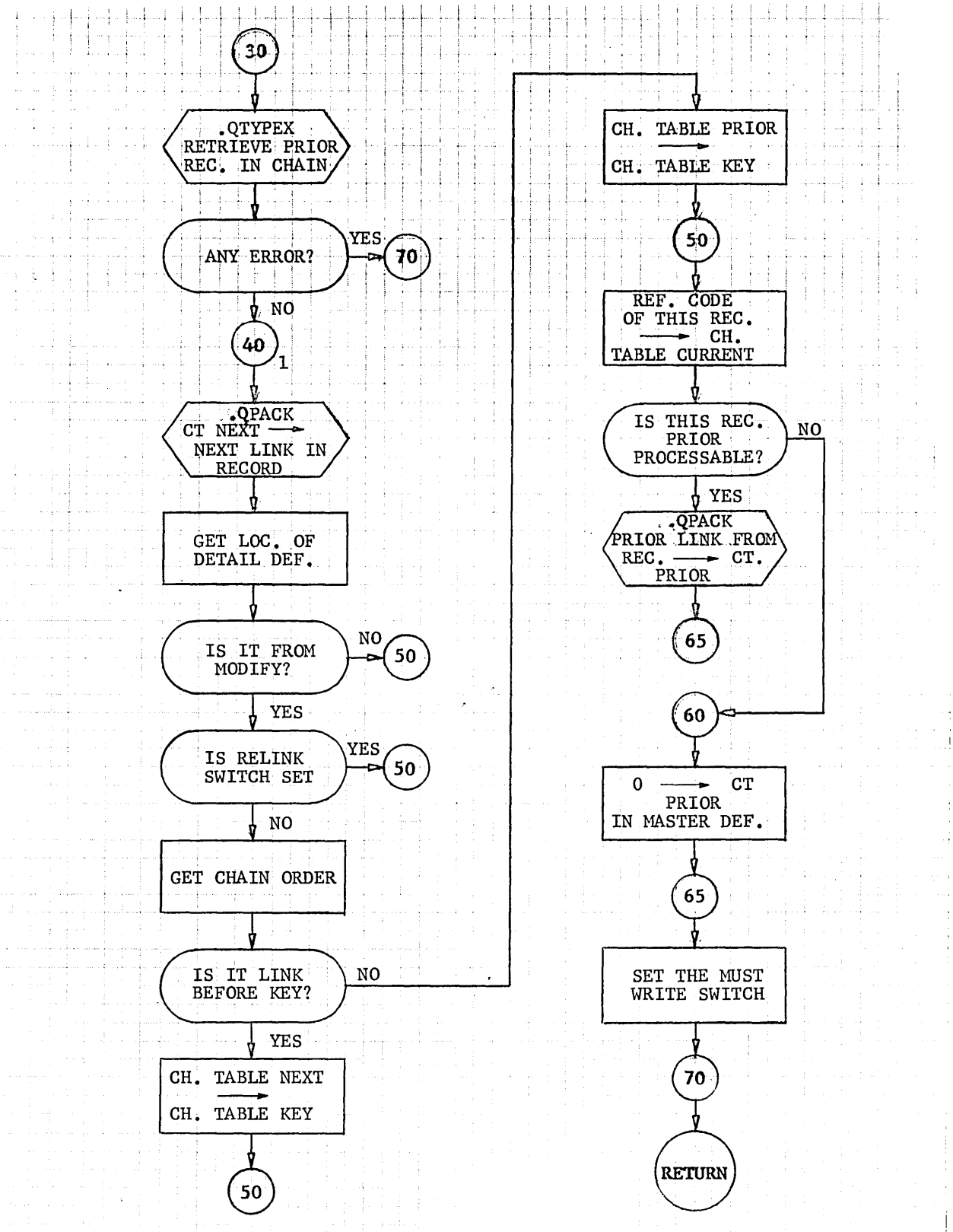
NUMBER OF WORDS:

205 octal

DETECTED ERRORS:

QUIT (57)	Attempt to delink a record when the chain table NEXT is equal to zero.


.QDLNK
SAVE LOC. OF CURRENT DET. DEF.
HEAD THE .CHAIN CHAIN
SAVE LOC. OF CURRENT MASTER DEF.
.QRUND
ANY ERROR?
YES
70
NO
IS MASTER REC. PRIOR PROCESSABLE?
NO
30
YES
IS CH. TABLE NEXT ZERO?
YES
ABORT QUIT 57
NO
10
10
.QTYPEX RETRIEVE NEXT REC.IN CHAIN
ANY ERROR?
YES
70
NO
IS THIS REC. PRIOR PROCESSABLE?
NO
YES
.QPACK CT PRIOR → PRIOR LINK IN REC.
IS CH. TABLE PRIOR = TO CH. TABLE NEXT?
YES
40
NO
SET MUST WRITE SWITCH
20
GET THE CURRENT MASTER DEF.
30


30
.QTYPEX
RETRIEVE PRIOR
REC. IN CHAIN
ANY ERROR?
YES
70
NO
40
1
.QPACK
CT NEXT → NEXT LINK IN
RECORD
GET LOC. OF
DETAIL DEF.
IS IT FROM
MODIFY?
NO
50
YES
IS RELINK
SWITCH SET
YES
50
NO
GET CHAIN ORDER
IS IT LINK
BEFORE KEY?
NO
YES
CH. TABLE NEXT
→
CH. TABLE KEY
50
CH. TABLE PRIOR
→
CH. TABLE KEY
50
REF. CODE
OF THIS REC.
→ CH.
TABLE CURRENT
IS THIS REC.
PRIOR
PROCESSABLE?
NO
YES
.QPACK
PRIOR LINK FROM
REC. → CT.
PRIOR
65
60
0 → CT
PRIOR
IN MASTER DEF.
65
SET THE MUST
WRITE SWITCH
70
RETURN

.QDLTE

FUNCTION:

Subroutine .QDLTE removes from the IDS file the last record retrieved and all its dependent details. If a conditional action list is specified, the deletion process is interrupted each time a dependent detail of a specified type is encountered, and the appropriate action is performed prior to the actual deletion of the record.

CALLING SEQUENCE:

CALL .QDLTE(ARG-1)

where ARG-1 is the location of the first word of a conditional action list or is zero.

CALLED BY:

Source program

CALLS FOR:

.QAUTH, .QUDCH, .QFWD, .QUPDC, .QMOVE, .QHEAD, .QDLNK, .QMRAC, .QSDSW, .QPACK, .QADJU, .QUIT

LIMITATIONS AND RESTRICTIONS:

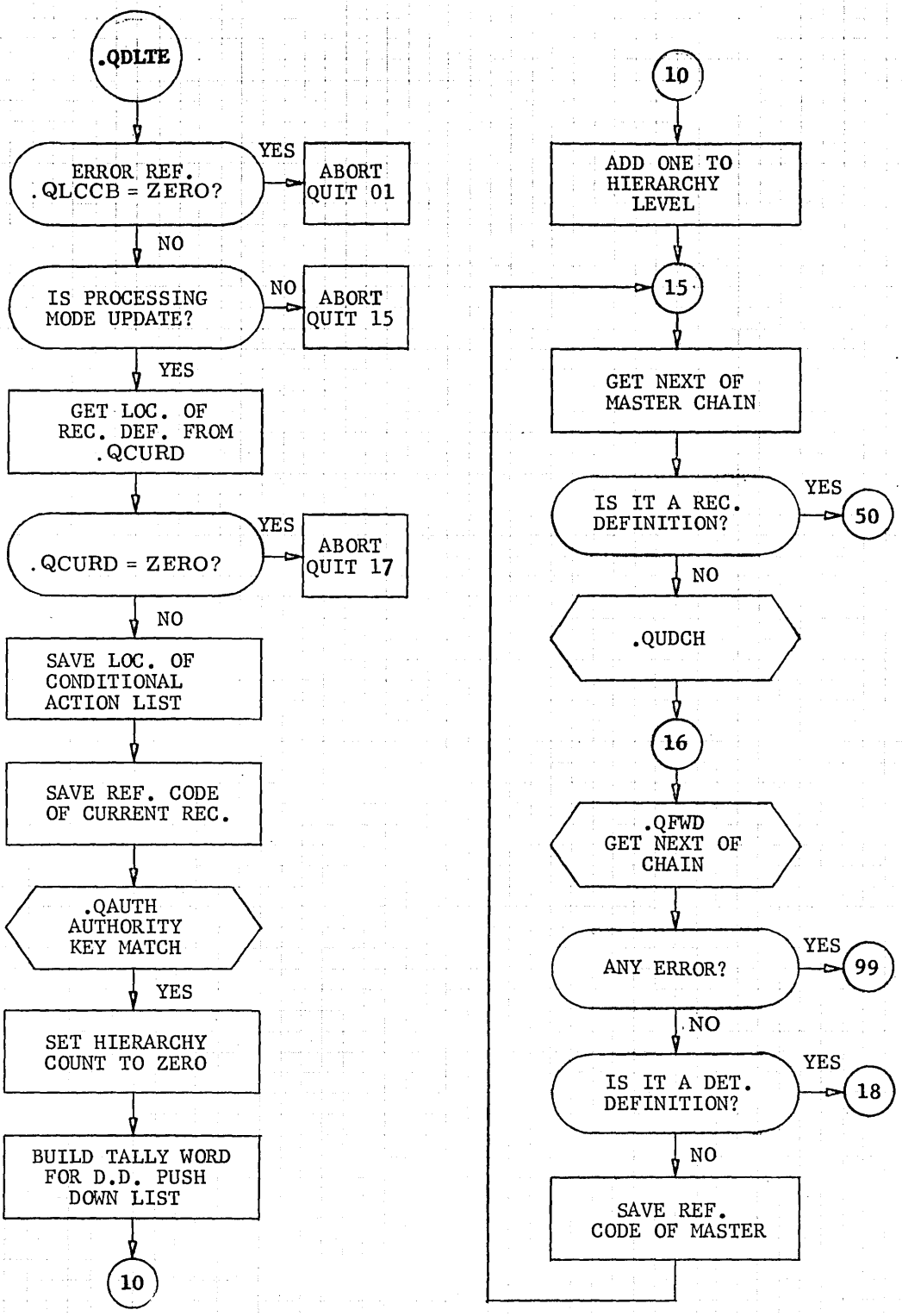
This subroutine checks the contents of error-reference to determine whether any error occurred during or since the retrieval of the record involved. If the contents are nonzero, the function of the routine is aborted; and an immediate return to the instruction following the call is executed.

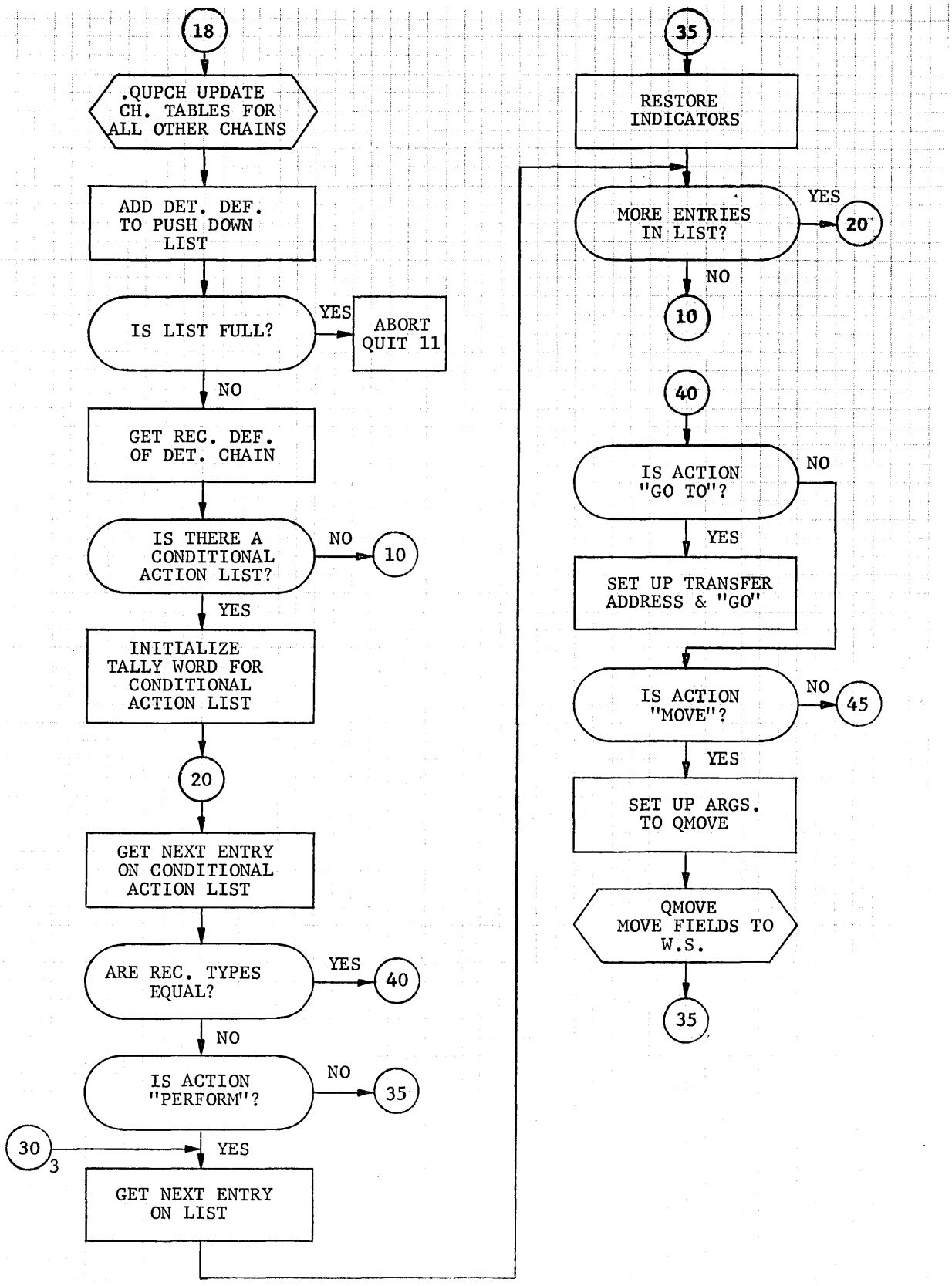
NUMBER OF WORDS:

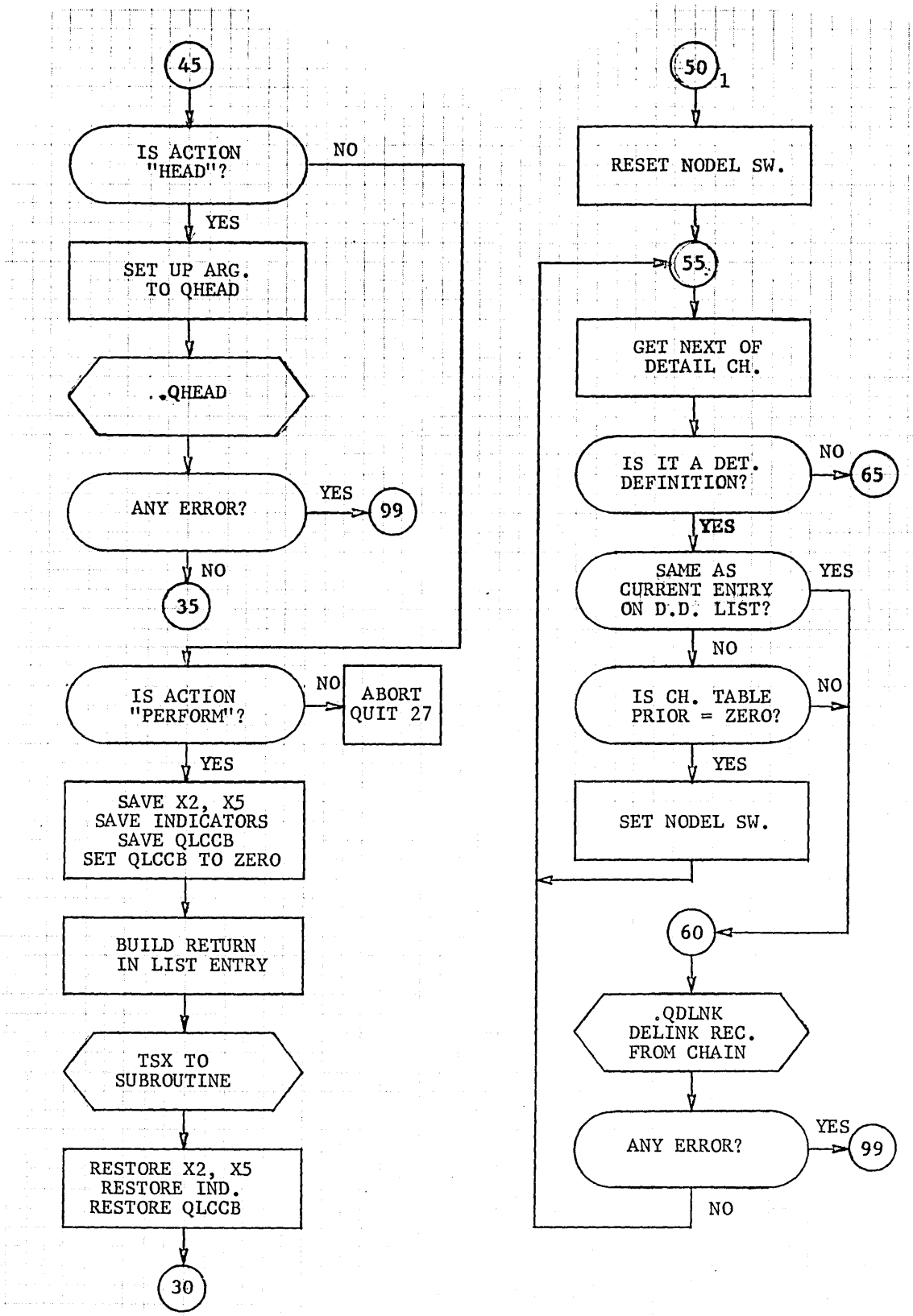
404 octal.

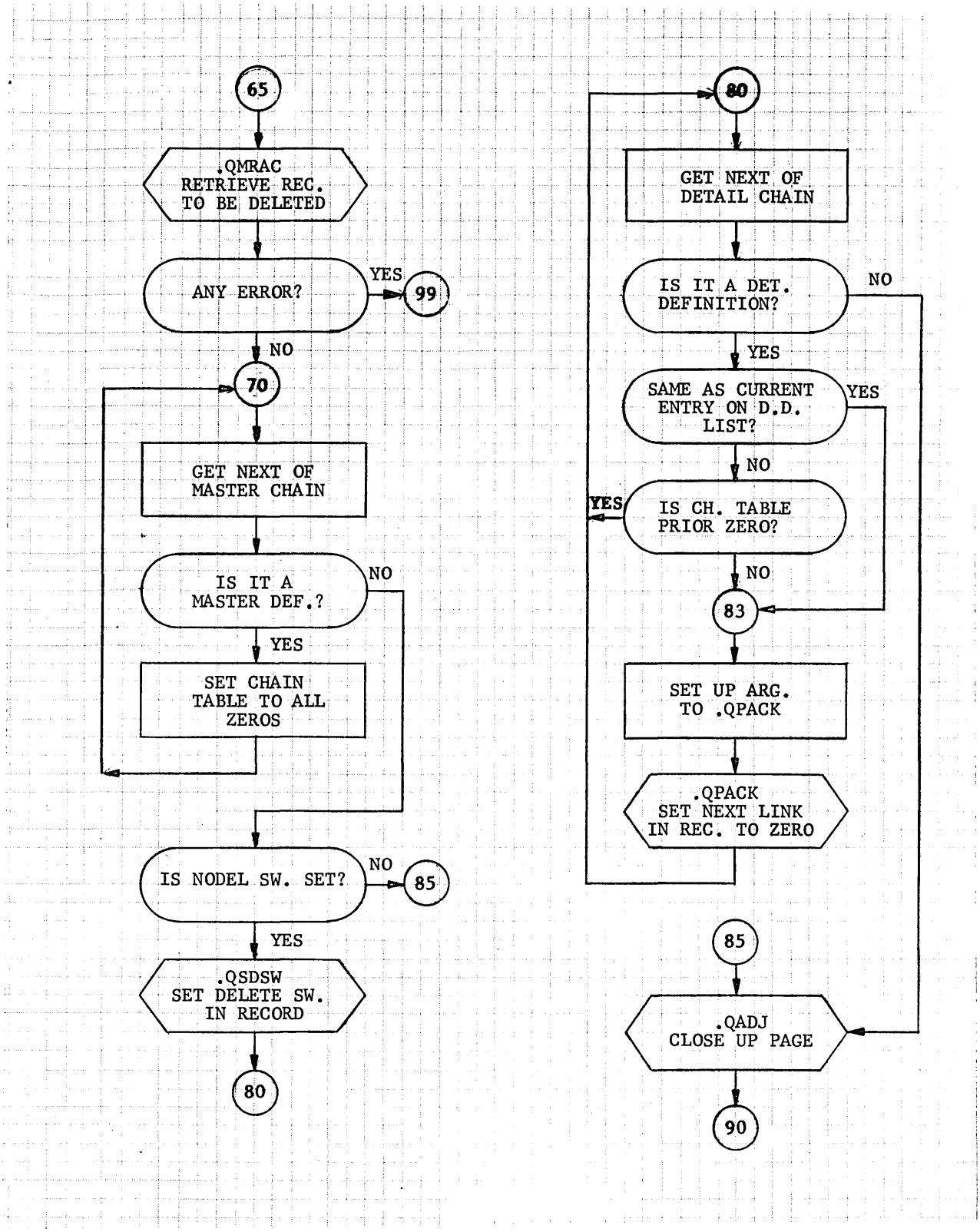
DETECTED ERRORS:

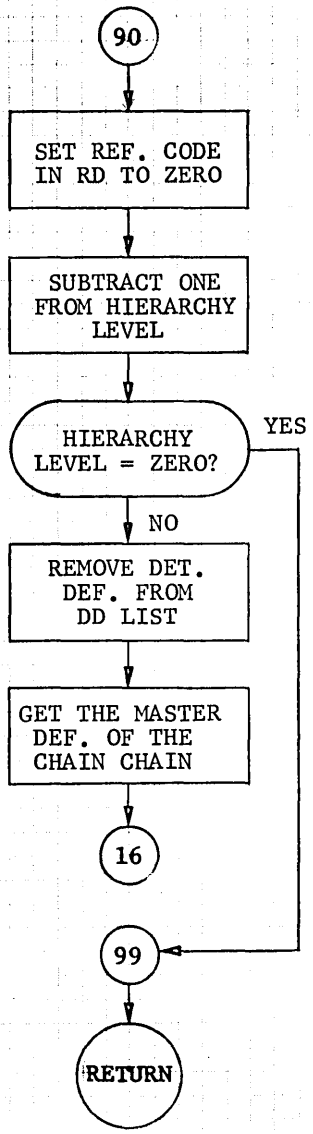
QUIT (01)	An error occurred before this routine was entered.
QUIT (15)	Attempt to modify, delete, or store a record, when the processing mode is not equal to update.
QUIT (17)	Attempt to delete a record not previously retrieved.
QUIT (11)	IDS subroutine table overflow. .QASC, .QDLTE, and .QSTOR must be reassembled.
QUIT (27)	Invalid conditional action specified for a conditional delete statement.


.QDLTE
ERROR REF. .QLCCB = ZERO?
YES
ABORT QUIT 01
NO
IS PROCESSING MODE UPDATE?
NO
ABORT QUIT 15
YES
GET LOC. OF REC. DEF. FROM .QCURD
.QCURD = ZERO?
YES
ABORT QUIT 17
NO
SAVE LOC. OF CONDITIONAL ACTION LIST
SAVE REF. CODE OF CURRENT REC.
.QAUTH AUTHORITY KEY MATCH
YES
SET HIERARCHY COUNT TO ZERO
BUILD TALLY WORD FOR D.D. PUSH DOWN LIST
10
10
ADD ONE TO HIERARCHY LEVEL
15
GET NEXT OF MASTER CHAIN
IS IT A REC. DEFINITION?
YES
50
NO
.QUDCH
16
.QFWD GET NEXT OF CHAIN
ANY ERROR?
YES
99
NO
IS IT A DET. DEFINITION?
YES
18
NO
SAVE REF. CODE OF MASTER


18
.QUPCH UPDATE CH. TABLES FOR ALL OTHER CHAINS
ADD DET. DEF. TO PUSH DOWN LIST
IS LIST FULL?
YES
ABORT QUIT 11
NO
GET REC. DEF. OF DET. CHAIN
IS THERE A CONDITIONAL ACTION LIST?
NO
10
YES
INITIALIZE TALLY WORD FOR CONDITIONAL ACTION LIST
20
GET NEXT ENTRY ON CONDITIONAL ACTION LIST
ARE REC. TYPES EQUAL?
YES
40
NO
IS ACTION "PERFORM"?
NO
35
30
3
YES
GET NEXT ENTRY ON LIST
35
RESTORE INDICATORS
MORE ENTRIES IN LIST?
YES
20
NO
10
40
IS ACTION "GO TO"?
NO
YES
SET UP TRANSFER ADDRESS & "GO"
IS ACTION "MOVE"?
NO
45
YES
SET UP ARGS. TO QMOVE
QMOVE MOVE FIELDS TO W.S.
35


45
IS ACTION "HEAD"?
NO
YES
SET UP ARG. TO QHEAD
..QHEAD
ANY ERROR?
YES
99
NO
35
IS ACTION "PERFORM"?
NO
ABORT QUIT 27
YES
SAVE X2, X5
SAVE INDICATORS
SAVE QLCCB
SET QLCCB TO ZERO
BUILD RETURN IN LIST ENTRY
TSX TO SUBROUTINE
RESTORE X2, X5
RESTORE IND.
RESTORE QLCCB
30
50
1
RESET NODEL SW.
55
GET NEXT OF DETAIL CH.
IS IT A DET. DEFINITION?
NO
65
YES
SAME AS CURRENT ENTRY ON D.D. LIST?
YES
NO
IS CH. TABLE PRIOR = ZERO?
NO
YES
SET NODEL SW.
60
.QDLNK
DELINK REC.
FROM CHAIN
ANY ERROR?
YES
99
NO


65
.QMRAC
RETRIEVE REC.
TO BE DELETED
ANY ERROR?
YES
99
NO
70
GET NEXT OF
MASTER CHAIN
IS IT A
MASTER DEF.?
NO
YES
SET CHAIN
TABLE TO ALL
ZEROS
IS NODEL SW. SET?
NO
85
YES
.QSDSW
SET DELETE SW.
IN RECORD
80
80
GET NEXT OF
DETAIL CHAIN
IS IT A DET.
DEFINITION?
NO
YES
SAME AS CURRENT
ENTRY ON D.D.
LIST?
YES
NO
YES
IS CH. TABLE
PRIOR ZERO?
NO
83
SET UP ARG.
TO .QPACK
.QPACK
SET NEXT LINK
IN REC. TO ZERO
85
.QADJ
CLOSE UP PAGE
90


90
SET REF. CODE IN RD TO ZERO
SUBTRACT ONE FROM HIERARCHY LEVEL
HIERARCHY LEVEL = ZERO?
YES
NO
REMOVE DET. DEF. FROM DD LIST
GET THE MASTER DEF. OF THE CHAIN CHAIN
16
99
RETURN

.QFWD

FUNCTION:

Subroutine .QFWD retrieves the next, prior, or master record of the chain specified. The chain table is updated, so that the record retrieved becomes the current record of the chain. Any record encountered with its delete switch is delinked from the chain.

CALLING SEQUENCE:

CALL .QFWD(ARG-1)

where ARG-1 is a decimal value indicating which record of the chain is to be retrieved:

1 = Next
2 = Prior
3 = Master

CALLED BY:

.QCHN, .QGDET, .QTLNK, .QDLTE

CALLS FOR:

.QRUND, .QTYPX, .QUDCH, .QDLNK, .QPACK, .QADJU, .QMRAC, .QUIT

LIMITATIONS AND RESTRICTIONS:

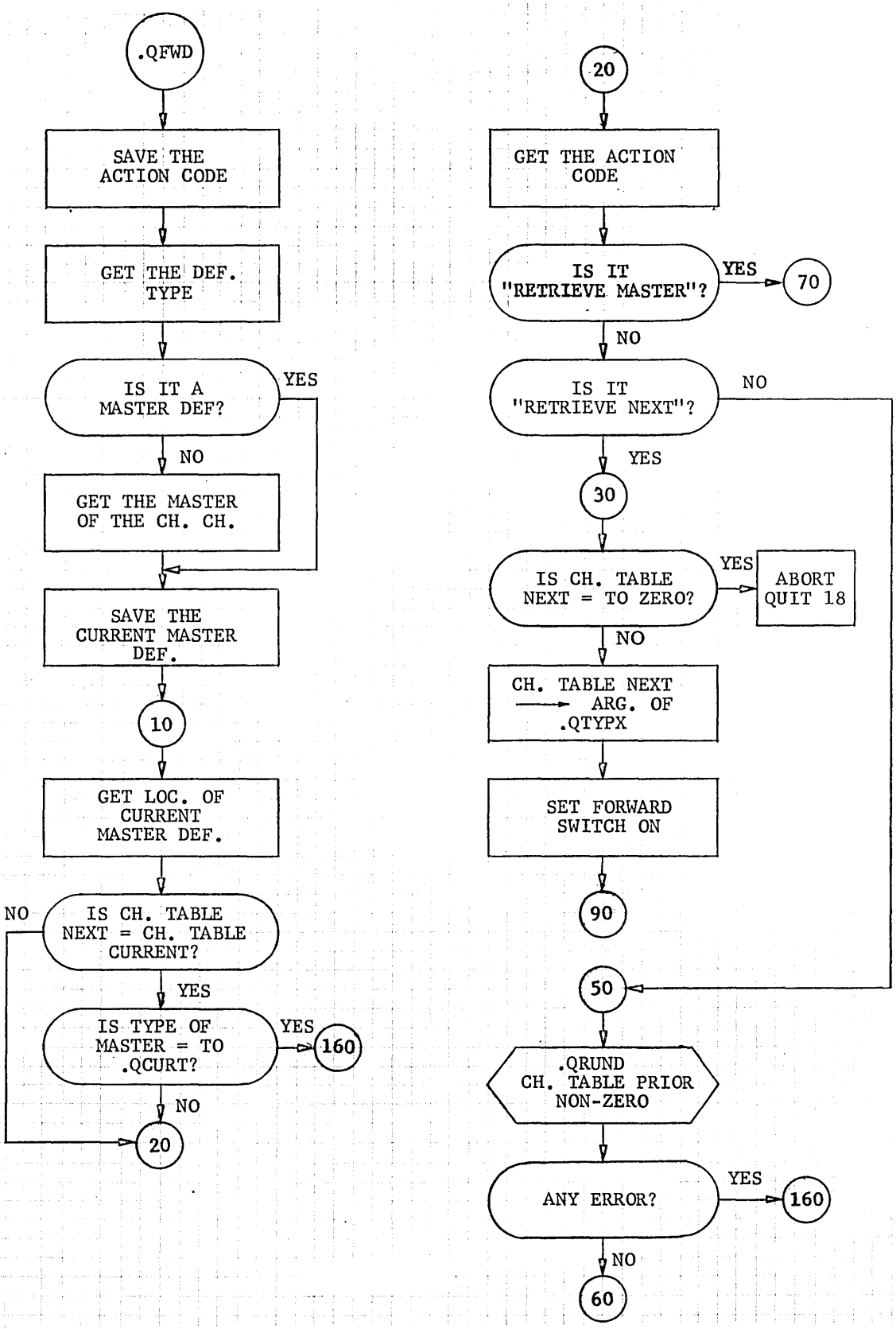
Index register 2 at entry contains the location of either a master definition or a detail definition designating the chain to be followed. At the normal exit, index register 2 contains the location of the master definition, if the master record of the chain was retrieved; it contains the detail definition, if a detail record of the chain was retrieved.

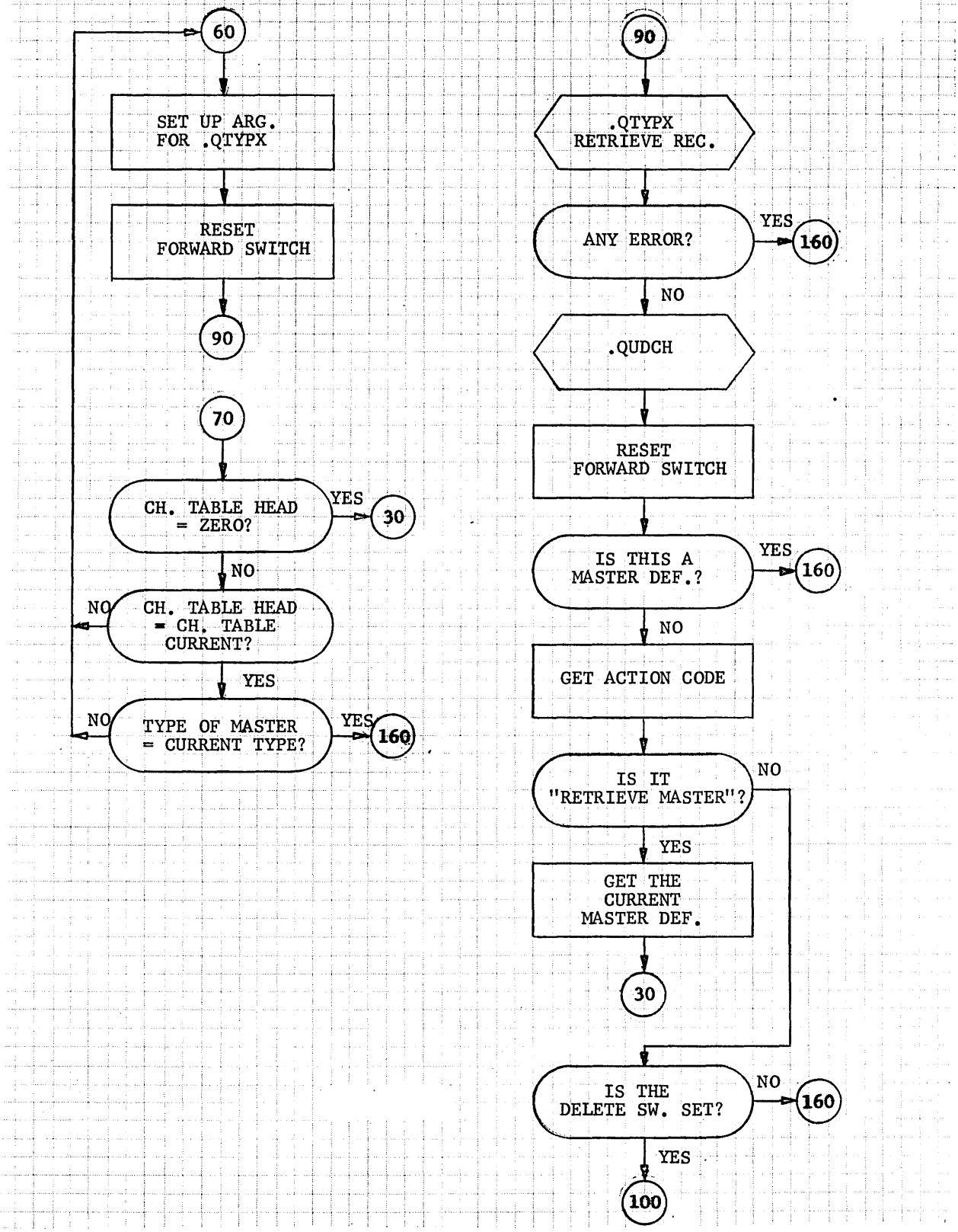
NUMBER OF WORDS:

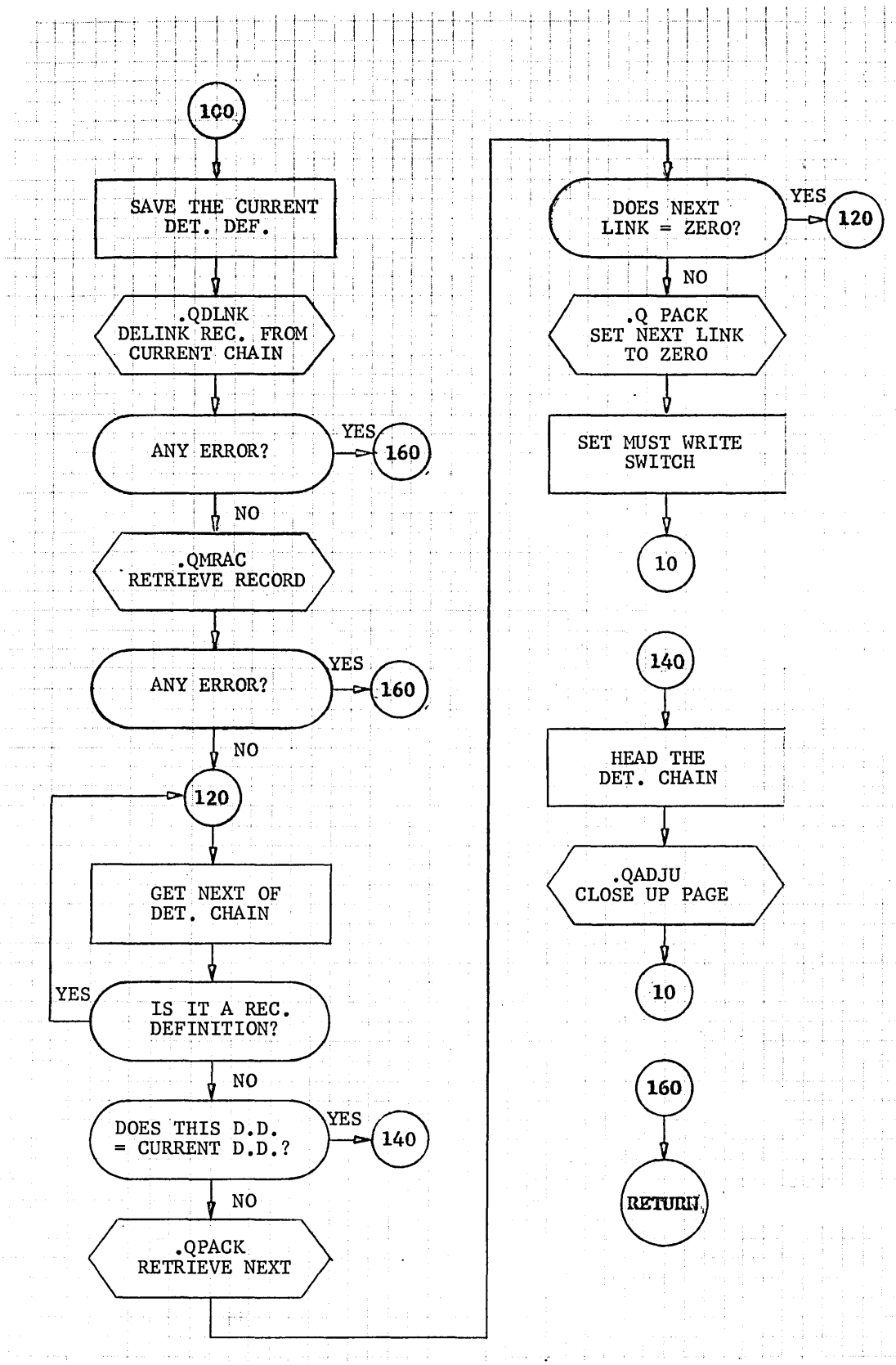
225 octal

DETECTED ERRORS:

QUIT (18) Attempt to retrieve next in a chain for which no current record exists.


.QFWD
SAVE THE ACTION CODE
GET THE DEF. TYPE
IS IT A MASTER DEF?
YES
NO
GET THE MASTER OF THE CH. CH.
SAVE THE CURRENT MASTER DEF.
10
GET LOC. OF CURRENT MASTER DEF.
NO
IS CH. TABLE NEXT = CH. TABLE CURRENT?
YES
IS TYPE OF MASTER = TO .QCURT?
YES
160
NO
20
20
GET THE ACTION CODE
IS IT "RETRIEVE MASTER"?
YES
70
NO
IS IT "RETRIEVE NEXT"?
NO
YES
30
IS CH. TABLE NEXT = TO ZERO?
YES
ABORT QUIT 18
NO
CH. TABLE NEXT → ARG. OF .QTYPX
SET FORWARD SWITCH ON
90
50
.QRUND CH. TABLE PRIOR NON-ZERO
ANY ERROR?
YES
160
NO
60


60
SET UP ARG. FOR .QTYPX
RESET FORWARD SWITCH
90
70
CH. TABLE HEAD = ZERO?
YES
30
NO
NO
CH. TABLE HEAD = CH. TABLE CURRENT?
YES
NO
TYPE OF MASTER = CURRENT TYPE?
YES
160
90
.QTYPX RETRIEVE REC.
ANY ERROR?
YES
160
NO
.QUDCH
RESET FORWARD SWITCH
IS THIS A MASTER DEF.?
YES
160
NO
GET ACTION CODE
IS IT "RETRIEVE MASTER"?
NO
YES
GET THE CURRENT MASTER DEF.
30
IS THE DELETE SW. SET?
NO
160
YES
100


100
SAVE THE CURRENT DET. DEF.
.QDLNK DELINK REC. FROM CURRENT CHAIN
ANY ERROR?
YES
160
NO
.QMRAC RETRIEVE RECORD
ANY ERROR?
YES
160
NO
120
GET NEXT OF DET. CHAIN
YES
IS IT A REC. DEFINITION?
NO
DOES THIS D.D. = CURRENT D.D.?
YES
140
NO
.QPACK RETRIEVE NEXT
DOES NEXT LINK = ZERO?
YES
120
NO
.Q PACK SET NEXT LINK TO ZERO
SET MUST WRITE SWITCH
10
140
HEAD THE DET. CHAIN
.QADJU CLOSE UP PAGE
10
160
RETURN

.QGDET

.QGDET uses a call to the library for a routine (.CNFXI) to convert any signed noncomputational fields to binary form.

FUNCTION:

Subroutine .QGDET retrieves the detail record whose control fields correspond to the values in working storage; if the detail record is not present, .QGDET retrieves its successor.

CALLING SEQUENCE:

CALL .QGDET(ARG-1, ARG-2)

where ARG-1 = 0, to retrieve a record
ARG-1 $\neq$ 0, to find insert point
ARG-2 contains the location of the detail definition of the required record, in bits 0-17.

CALLED BY:

.QASC, .QTLNK

CALLS FOR:

.QFWD, .CNFXI, .QPACK, .QUIT

LIMITATIONS AND RESTRICTIONS:

Index register 2 at entry contains the location of either a master definition or a detail definition designating the chain to which the desired record belongs.

NUMBER OF WORDS:

222 octal.

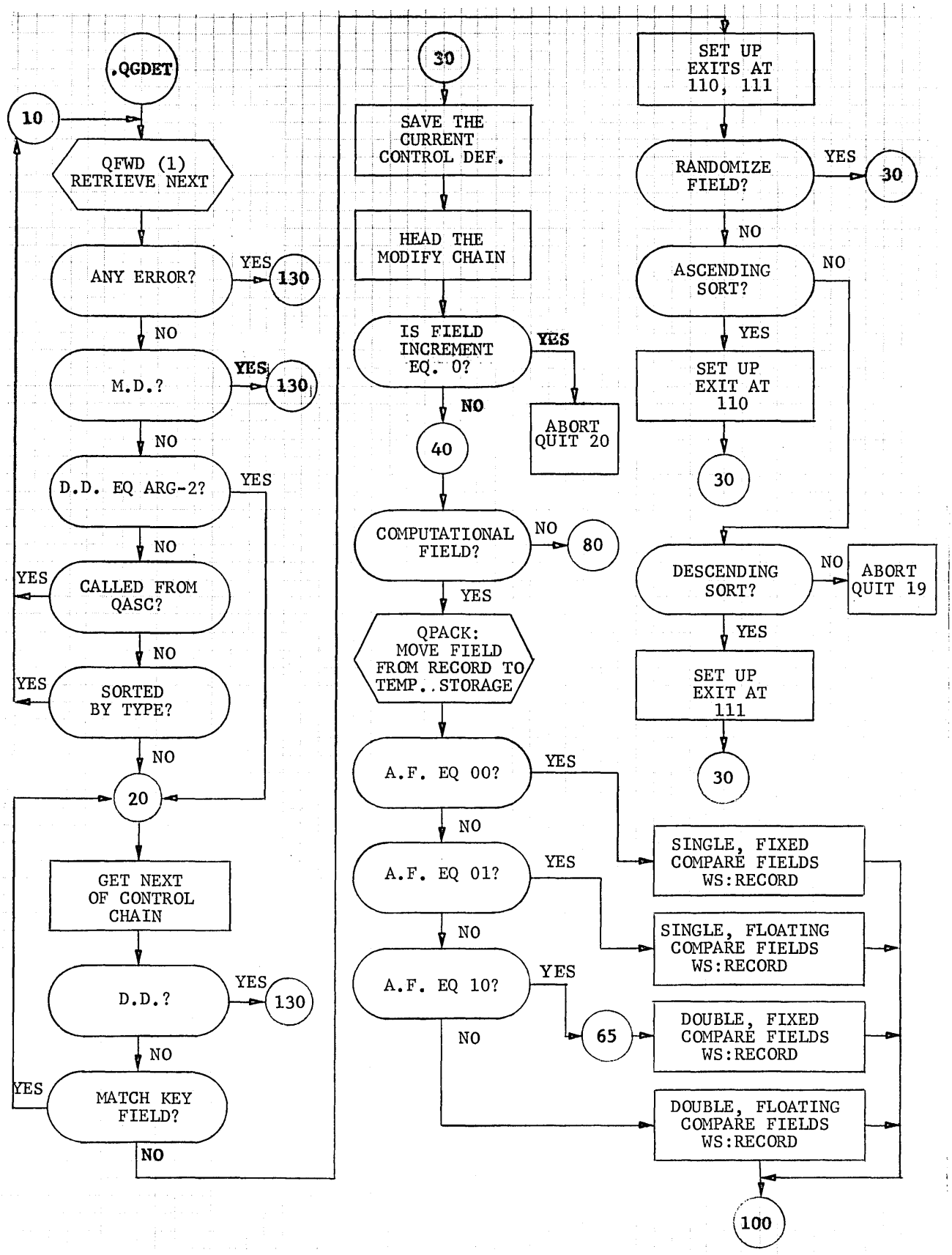
RETURNS TO .QGDET:

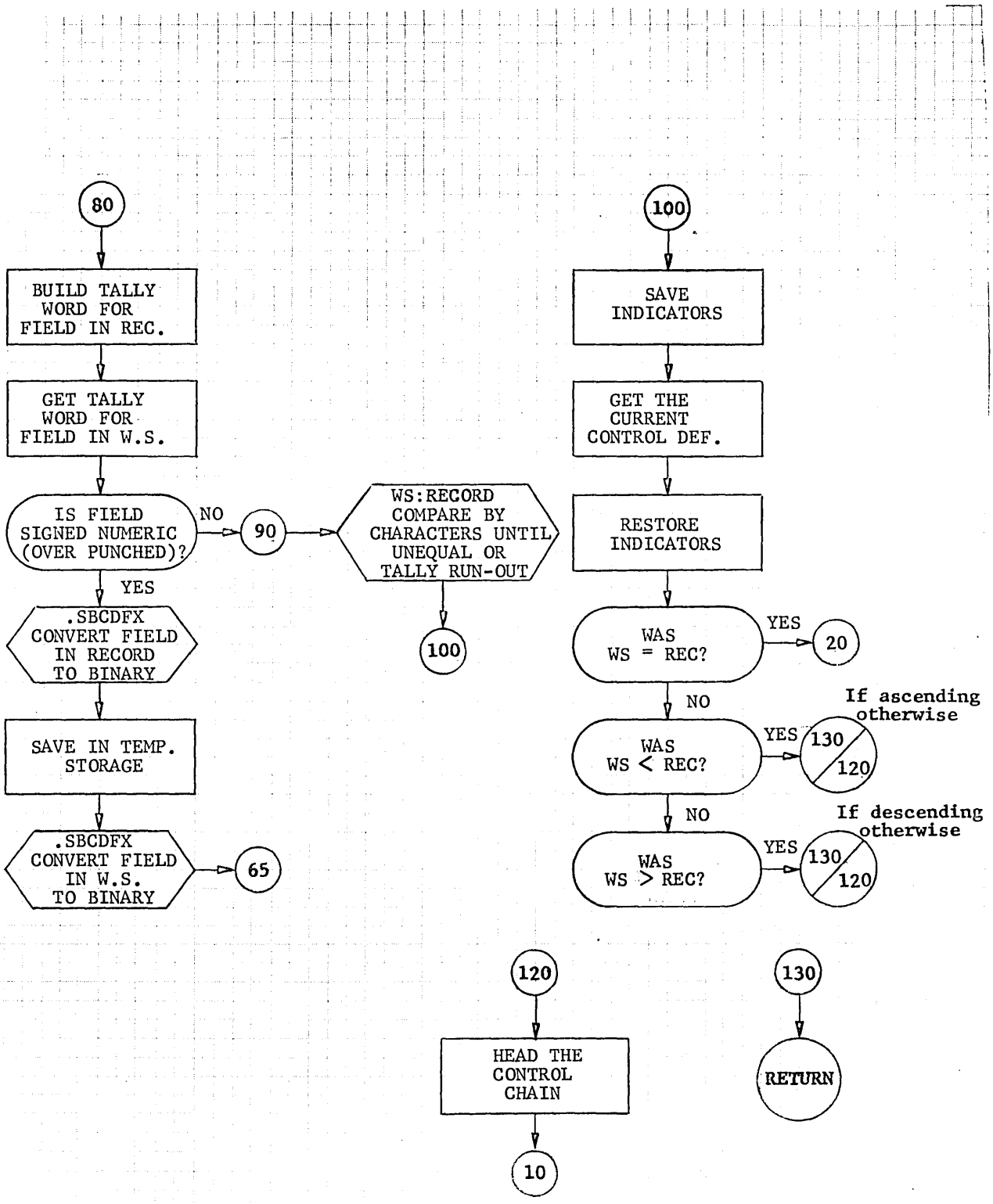
	From .QASC	From .QLNK
Error	Error someplace	Error someplace
D.D in X2	Retrieved record with all control fields equal.	Retrieved a duplicate.
M.D in X2	Error: valid detail not present in chain.	Store new record before master; that is, last in chain.
C.D in X2	Retrieved a record with control fields greater than working storage; may be acceptable in case of range masters.	Retrieved record just after insert point of new record; store new record before this one.

ERRORS DETECTED:

QUIT (19) A control definition has been encountered which has an invalid control code.

QUIT (20) A sort control or a randomize control field is specified with a record increment of zero.


.QGDET
10
QFWD (1) RETRIEVE NEXT
ANY ERROR?
YES
130
NO
M.D.?
YES
130
NO
D.D. EQ ARG-2?
YES
NO
CALLED FROM QASC?
YES
NO
SORTED BY TYPE?
YES
NO
20
GET NEXT OF CONTROL CHAIN
D.D.?
YES
130
NO
MATCH KEY FIELD?
YES
NO
30
SAVE THE CURRENT CONTROL DEF.
HEAD THE MODIFY CHAIN
IS FIELD INCREMENT EQ. 0?
YES
ABORT QUIT 20
NO
40
COMPUTATIONAL FIELD?
NO
80
YES
QPACK: MOVE FIELD FROM RECORD TO TEMP..STORAGE
A.F. EQ 00?
YES
NO
A.F. EQ 01?
YES
NO
A.F. EQ 10?
YES
65
NO
SET UP EXITS AT 110, 111
RANDOMIZE FIELD?
YES
30
NO
ASCENDING SORT?
NO
YES
SET UP EXIT AT 110
30
DESCENDING SORT?
NO
ABORT QUIT 19
YES
SET UP EXIT AT 111
30
SINGLE, FIXED COMPARE FIELDS WS:RECORD
SINGLE, FLOATING COMPARE FIELDS WS:RECORD
DOUBLE, FIXED COMPARE FIELDS WS:RECORD
DOUBLE, FLOATING COMPARE FIELDS WS:RECORD
100


80
BUILD TALLY WORD FOR FIELD IN REC.
GET TALLY WORD FOR FIELD IN W.S.
IS FIELD SIGNED NUMERIC (OVER PUNCHED)?
NO
90
WS:RECORD COMPARE BY CHARACTERS UNTIL UNEQUAL OR TALLY RUN-OUT
100
YES
.SBCDFX CONVERT FIELD IN RECORD TO BINARY
SAVE IN TEMP. STORAGE
.SBCDFX CONVERT FIELD IN W.S. TO BINARY
65
100
SAVE INDICATORS
GET THE CURRENT CONTROL DEF.
RESTORE INDICATORS
WAS WS = REC?
YES
20
NO
If ascending otherwise
WAS WS < REC?
YES
130
120
NO
If descending otherwise
WAS WS > REC?
YES
130
120
120
HEAD THE CONTROL CHAIN
10
130
RETURN

.QGET

FUNCTION:

Subroutine .QGET retrieves a record of the specified type in a manner defined by its data description. It retrieves a primary record by using the field which is equivalent to the reference code to locate the record. A secondary record is retrieved because of its association as a detail in its defined retrieval chain. In this case, the subroutine must first retrieve the master of the chain. Then, by searching the chain, it finds the record. .QGET retrieves a calculated record by using the randomize fields to find a page, and then by searching the CALC chain for that page. After it locates the record in one of the memory buffers, it stores the reference code and record type in the communication control block.

CALLING SEQUENCE:

CALL .QGET(ARG-1)

where ARG-1 is the location of the record definition for the record type to be retrieved.

CALLED BY:

Source program

CALLS FOR:

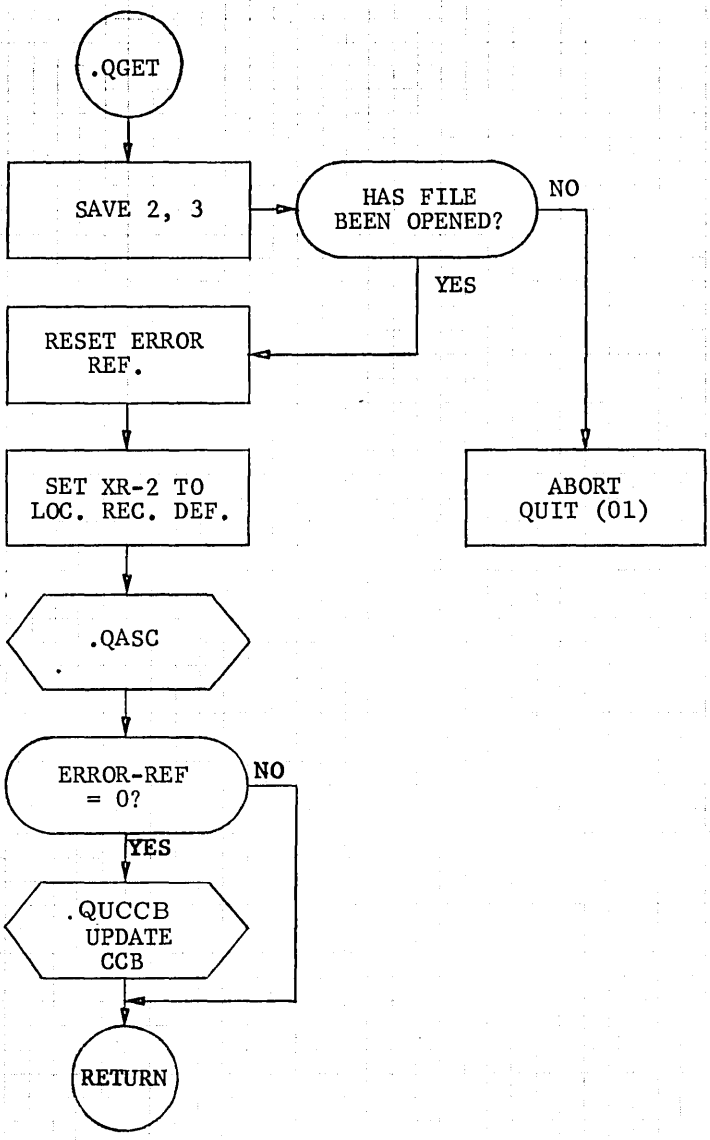
.QASC, .QUIT, .QUCCB

NUMBER OF WORDS:

40 octal.

DETECTED ERRORS:

QUIT (01) Attempt to store or retrieve a record with the IDS file not opened.


.QGET
SAVE 2, 3
HAS FILE BEEN OPENED?
NO
YES
RESET ERROR REF.
SET XR-2 TO LOC. REC. DEF.
ABORT QUIT (01)
.QASC
ERROR-REF = 0?
NO
YES
.QUCCB UPDATE CCB
RETURN

.QGETC

FUNCTION:

Subroutine .QGETC retrieves the last record of the type specified that was processed by any store or retrieve call. If the last record was deleted, an error condition is returned to the user via word 5 of the communication control block.

CALLING SEQUENCE:

CALL .QGETC(ARG-1)

where ARG-1 is the location of the record definition for the record type to be retrieved.

CALLED BY:

Source program

CALLS FOR:

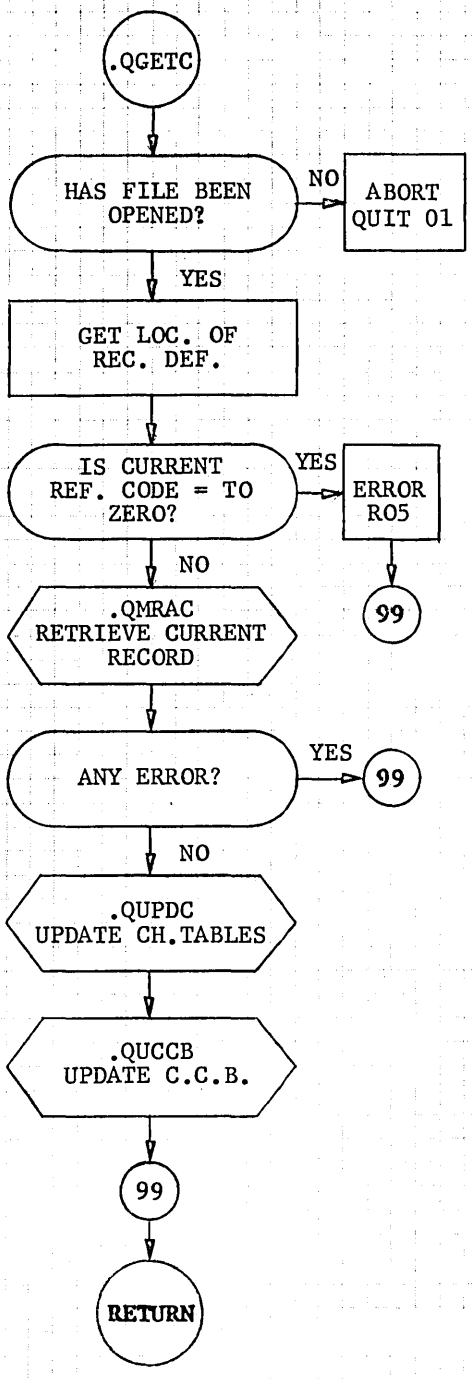
.QUIT, .QMRAC, .QUCCB, .QUPDC

NUMBER OF WORDS:

51 octal

DETECTED ERRORS:

QUIT (01)	Attempt to retrieve or store a record with the IDS file not opened.
R05	Attempt to retrieve the current record of a given type when that record type has not been previously retrieved or has been deleted.


.QGETC
HAS FILE BEEN OPENED?
NO
ABORT QUIT 01
YES
GET LOC. OF REC. DEF.
IS CURRENT REF. CODE = TO ZERO?
YES
ERROR R05
99
NO
.QMRAC RETRIEVE CURRENT RECORD
ANY ERROR?
YES
99
NO
.QUPDC UPDATE CH.TABLES
.QUCCB UPDATE C.C.B.
99
RETURN

.QGETD

FUNCTION:

Subroutine .QGETD retrieves the record identified by the reference code in the direct-reference field of the communication control block.

CALLING SEQUENCE:

```
CALL   .QGETD
```

CALLED BY:

Source program

CALLS FOR:

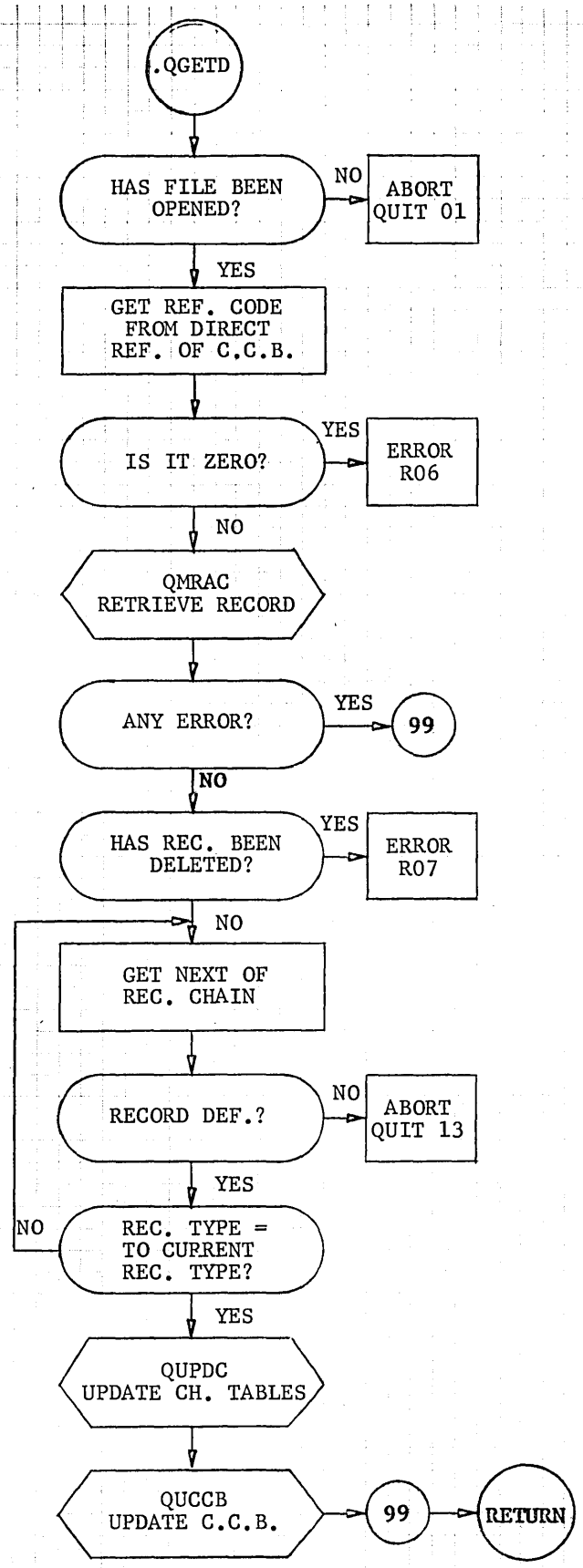
.QUIT, .QMRAC, .QUCCB, .QUPDC

NUMBER OF WORDS:

66 octal.

DETECTED ERRORS:

QUIT (01)	Attempt to retrieve or store a record with the IDS file not opened.
QUIT (13)	No record definition has been defined for this record type.
R06	Attempt to retrieve DIRECT when DIRECT-REFERENCE is zero.
R07	Attempt to retrieve a record which has been deleted and has its delete switch set.


.QGETD
HAS FILE BEEN OPENED?
NO
ABORT QUIT 01
YES
GET REF. CODE FROM DIRECT REF. OF C.C.B.
IS IT ZERO?
YES
ERROR R06
NO
QMRAC RETRIEVE RECORD
ANY ERROR?
YES
99
NO
HAS REC. BEEN DELETED?
YES
ERROR R07
NO
GET NEXT OF REC. CHAIN
RECORD DEF.?
NO
ABORT QUIT 13
YES
NO
REC. TYPE = TO CURRENT REC. TYPE?
YES
QUPDC UPDATE CH. TABLES
QUCCB UPDATE C.C.B.
99
RETURN

.QGETE

FUNCTION:

Subroutine .QGETE retrieves the first record found in the range of reference codes defined by the first-reference and last-reference fields of the communication control block. With each successful retrieval, the contents of first-reference are modified to the next-reference code in the range. When the reference code of first-reference is greater than that of last-reference, the exit specified by ARG-1 of the call is taken.

CALLING SEQUENCE:

CALL .QGETE(ARG-1)

where ARG-1 is the address of the "At End" exit.

CALLED BY:

Source program

CALLS FOR:

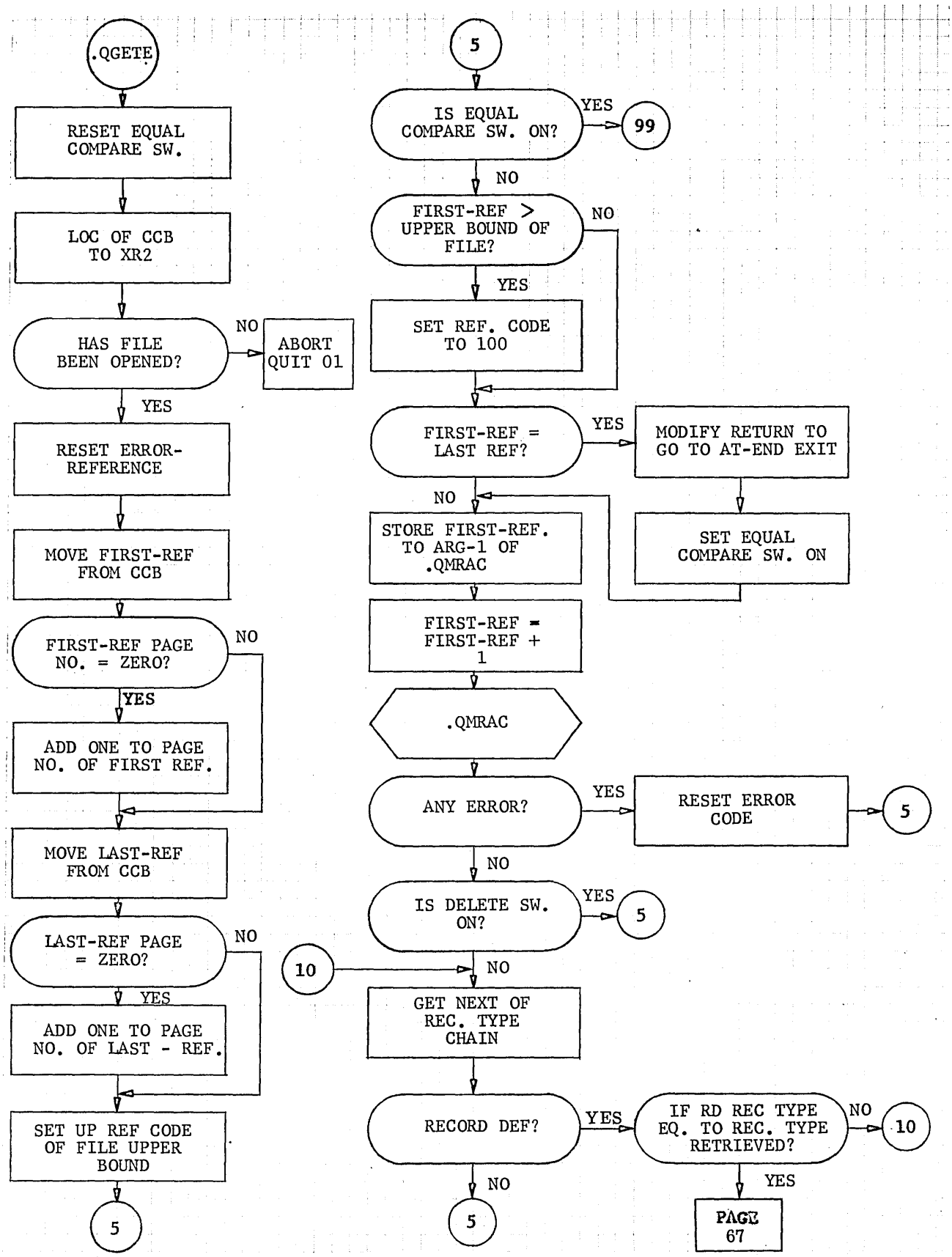
.QUIT, .QMRAC, .QUCCB, .QUPDC

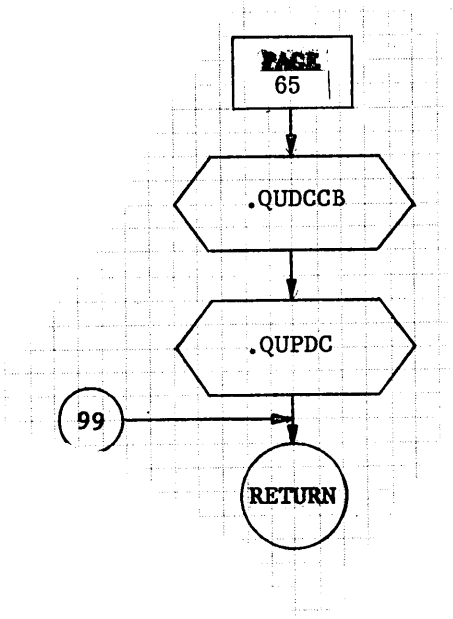
NUMBER OF WORDS:

117 octal

DETECTED ERRORS:

QUIT (01) Attempt to retrieve or store a record with the IDS file not opened.


.QGETE
RESET EQUAL COMPARE SW.
LOC OF CCB TO XR2
HAS FILE BEEN OPENED?
NO
ABORT QUIT 01
YES
RESET ERROR-REFERENCE
MOVE FIRST-REF FROM CCB
FIRST-REF PAGE NO. = ZERO?
NO
YES
ADD ONE TO PAGE NO. OF FIRST REF.
MOVE LAST-REF FROM CCB
LAST-REF PAGE = ZERO?
NO
YES
ADD ONE TO PAGE NO. OF LAST - REF.
SET UP REF CODE OF FILE UPPER BOUND
5
5
IS EQUAL COMPARE SW. ON?
YES
99
NO
FIRST-REF > UPPER BOUND OF FILE?
NO
YES
SET REF. CODE TO 100
FIRST-REF = LAST REF?
YES
MODIFY RETURN TO GO TO AT-END EXIT
NO
STORE FIRST-REF. TO ARG-1 OF .QMRAC
SET EQUAL COMPARE SW. ON
FIRST-REF = FIRST-REF + 1
.QMRAC
ANY ERROR?
YES
RESET ERROR CODE
5
NO
IS DELETE SW. ON?
YES
5
10
NO
GET NEXT OF REC. TYPE CHAIN
RECORD DEF?
YES
IF RD REC TYPE EQ. TO REC. TYPE RETRIEVED?
NO
10
NO
YES
5
PAGE 67


PAGE
65
.QUDCCB
.QUPDC
99
RETURN

.QHEAD

FUNCTION:

Subroutine .QHEAD retrieves the master record of the specified chain. The data fields of the master record are moved to working storage. Chain tables of chains in which this record is a detail are updated. This record becomes the CURRENT of its type.

CALLING SEQUENCE:

CALL .QHEAD(ARG-1)

where ARG-1 is the location of the master definition for the chain whose master record is to be retrieved.

CALLED BY:

Source program, .QDLTE

CALLS FOR:

.QTYPX, .QPACK, .QMOVE, .QUDCH, .QUIT

NUMBER OF WORDS:

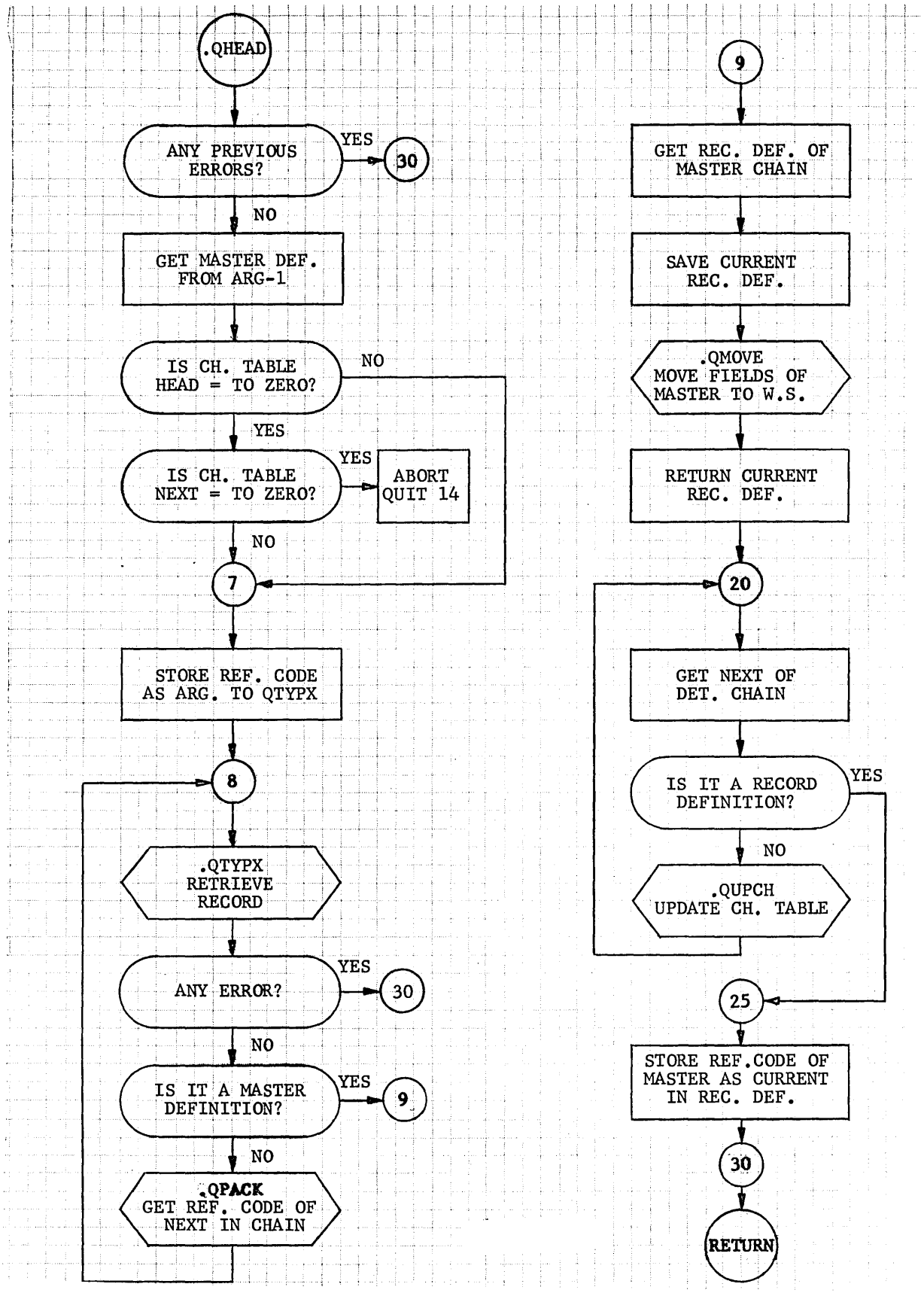
124 octal

LIMITATIONS AND RESTRICTIONS:

This routine checks the contents of ERROR-REFERENCE to determine whether any error occurred during or since the previous retrieval. If the contents are nonzero, the routine is not executed; and control is returned to the instruction following the call.

DETECTED ERRORS:

QUIT (14) Attempt to head chain for which no current record exists.


.QHEAD
ANY PREVIOUS ERRORS?
YES
30
NO
GET MASTER DEF. FROM ARG-1
IS CH. TABLE HEAD = TO ZERO?
NO
YES
IS CH. TABLE NEXT = TO ZERO?
YES
ABORT QUIT 14
NO
7
STORE REF. CODE AS ARG. TO QTYPX
8
.QTYPX RETRIEVE RECORD
ANY ERROR?
YES
30
NO
IS IT A MASTER DEFINITION?
YES
9
NO
.QPACK GET REF. CODE OF NEXT IN CHAIN
9
GET REC. DEF. OF MASTER CHAIN
SAVE CURRENT REC. DEF.
.QMOVE MOVE FIELDS OF MASTER TO W.S.
RETURN CURRENT REC. DEF.
20
GET NEXT OF DET. CHAIN
IS IT A RECORD DEFINITION?
YES
NO
.QUPCH UPDATE CH. TABLE
25
STORE REF.CODE OF MASTER AS CURRENT IN REC. DEF.
30
RETURN

.QLOCK

FUNCTION:

Subroutine .QLOCK locks a page in core by setting the entry in the activity chain corresponding to the buffer containing the page to -1 and delinking the entry from the activity chain. If the buffer is already locked, each further call reduces the value in the activity chain entry by 1.

CALLING SEQUENCE:

CALL .QLOCK(ARG-1)

where ARG-1 is a word containing the number of the buffer to be locked, in bits 0-17.

CALLED BY:

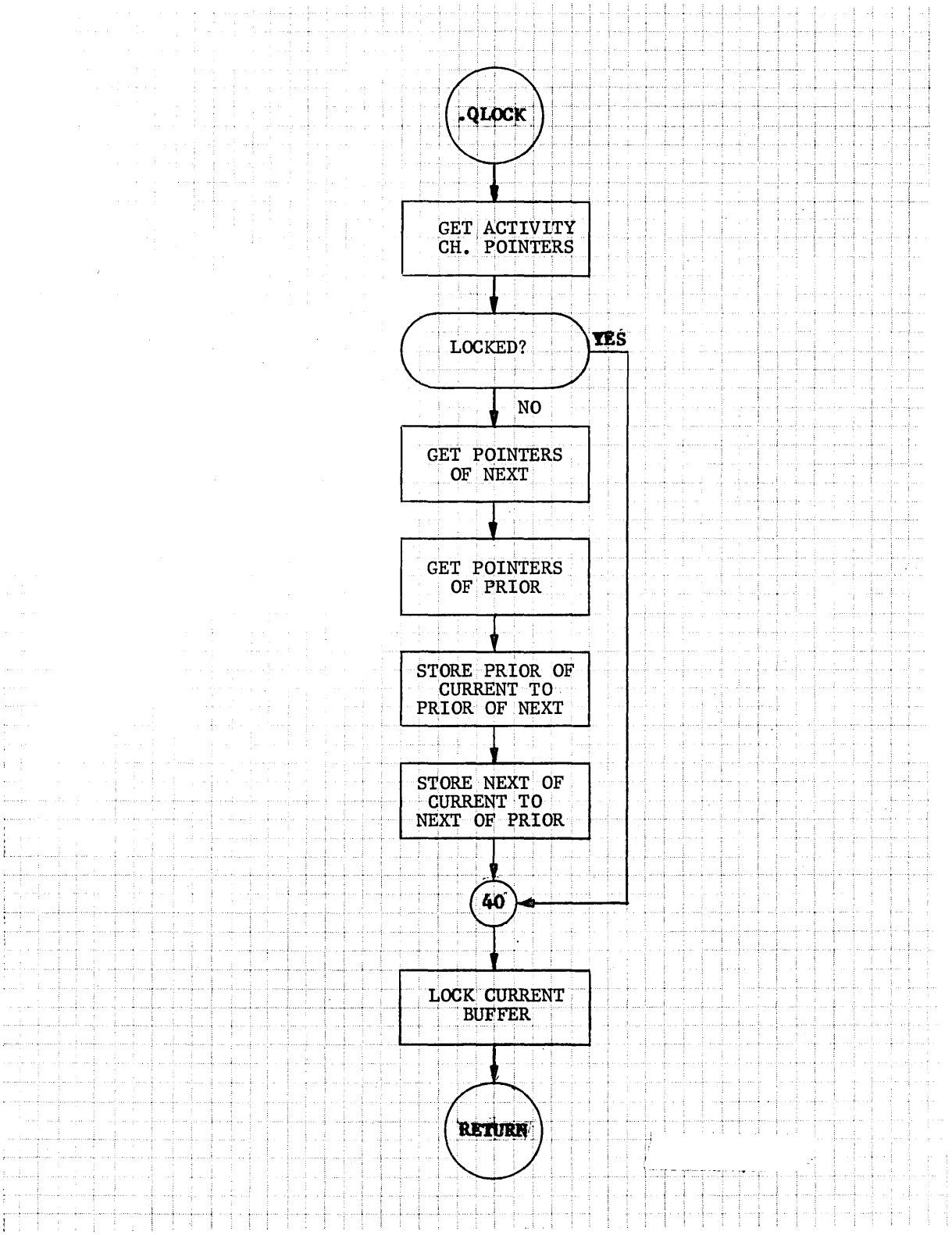
.QBIC, .QMNO

CALLS FOR:

Not applicable

NUMBER OF WORDS:

25 octal


.QLOCK
GET ACTIVITY CH. POINTERS
LOCKED?
YES
NO
GET POINTERS OF NEXT
GET POINTERS OF PRIOR
STORE PRIOR OF CURRENT TO PRIOR OF NEXT
STORE NEXT OF CURRENT TO NEXT OF PRIOR
40
LOCK CURRENT BUFFER
RETURN

.QMDFY

FUNCTION:

Subroutine .QMDFY replaces the contents of the specified fields of the record last retrieved with the contents of working storage. If a field to be modified is a match-key field, the record may be delinked from its old master, its new master retrieved, and the record properly linked to its new master. In this case, there is no change of the field; the record has become part of a different chain. If a field to be modified is a sort control field, the record is delinked from its master and relinked to it again according to the new value of the field.

CALLING SEQUENCE:

CALL .QMDFY(ARG-1, ARG-2)

where ARG-1 is the location of the first word of a list of working storage locations.

ARG-2 contains a binary integer indicating the length of the list.

CALLED BY:

Source program

CALLS FOR:

.QAUTH, .QPACK, .QTLNK, .QUPDC, .QUIT

NUMBER OF WORDS:

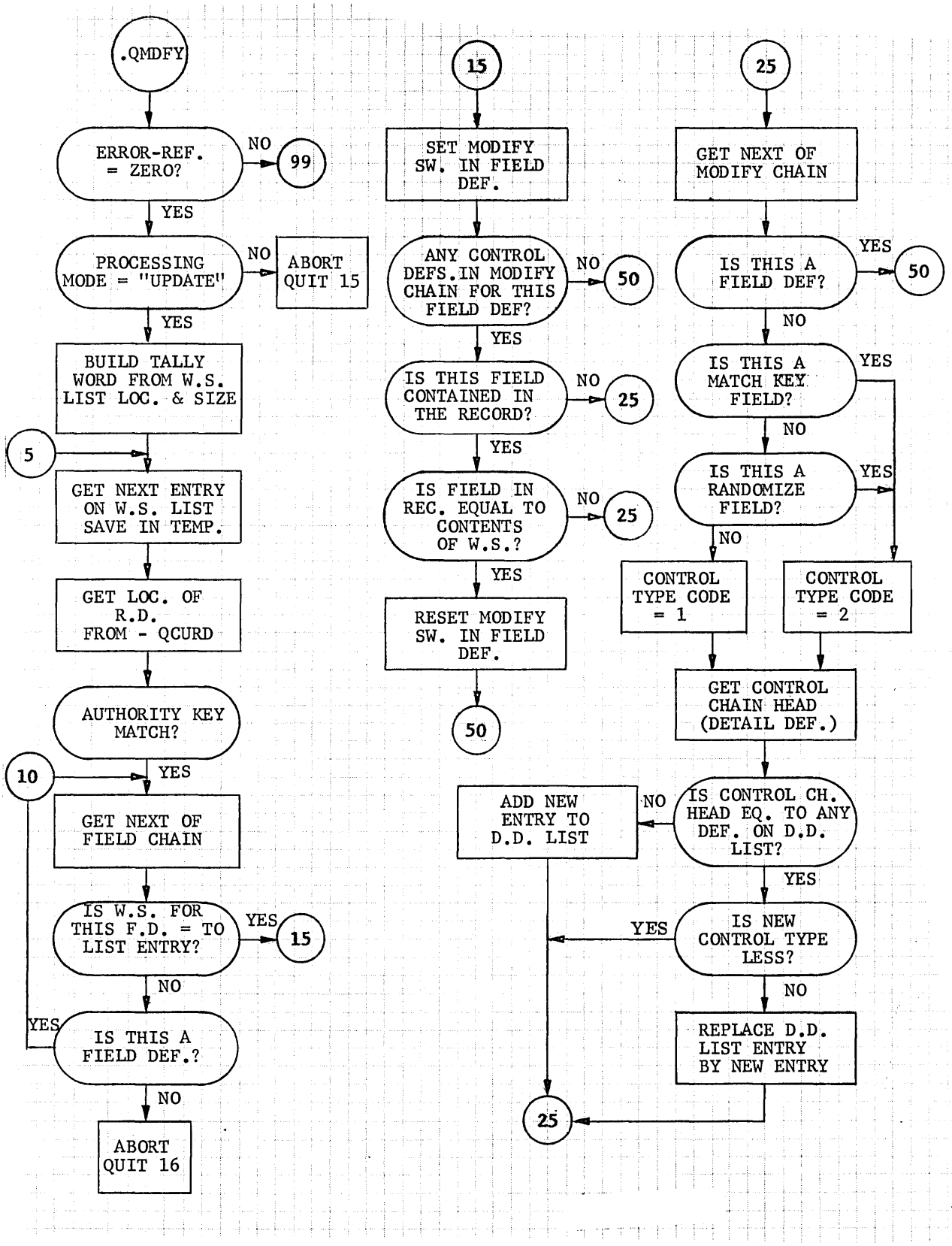
264 octal

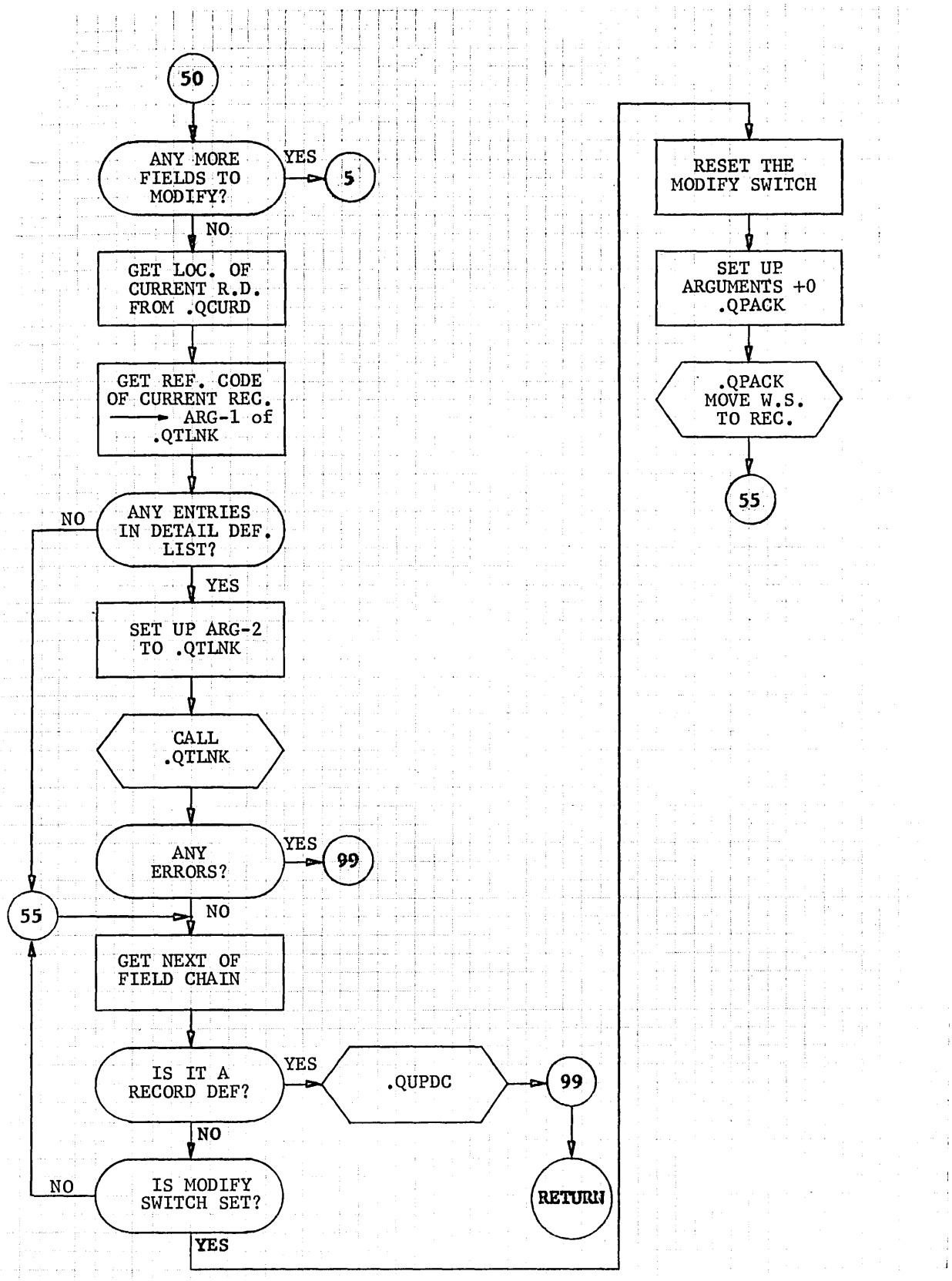
LIMITATIONS AND RESTRICTIONS:

This routine checks the contents of ERROR-REFERENCE to determine whether any error occurred during or since the retrieval of the record involved. If the contents are nonzero, the function of the routine is aborted; and an immediate return to the instruction following the call is executed.

DETECTED ERRORS:

QUIT (15)	Attempt to modify, delete, or store a record when the processing mode is not equal to update.
QUIT (16)	Field to be modified or moved not defined for the current record.


.QMDFY
ERROR-REF. = ZERO?
NO
99
YES
PROCESSING MODE = "UPDATE"
NO
ABORT QUIT 15
YES
BUILD TALLY WORD FROM W.S. LIST LOC. & SIZE
5
GET NEXT ENTRY ON W.S. LIST SAVE IN TEMP.
GET LOC. OF R.D. FROM - QCURD
AUTHORITY KEY MATCH?
10
YES
GET NEXT OF FIELD CHAIN
IS W.S. FOR THIS F.D. = TO LIST ENTRY?
YES
15
NO
YES
IS THIS A FIELD DEF.?
NO
ABORT QUIT 16
15
SET MODIFY SW. IN FIELD DEF.
ANY CONTROL DEFS.IN MODIFY CHAIN FOR THIS FIELD DEF?
NO
50
YES
IS THIS FIELD CONTAINED IN THE RECORD?
NO
25
YES
IS FIELD IN REC. EQUAL TO CONTENTS OF W.S.?
NO
25
YES
RESET MODIFY SW. IN FIELD DEF.
50
25
GET NEXT OF MODIFY CHAIN
IS THIS A FIELD DEF?
YES
50
NO
IS THIS A MATCH KEY FIELD?
YES
NO
IS THIS A RANDOMIZE FIELD?
YES
NO
CONTROL TYPE CODE = 1
CONTROL TYPE CODE = 2
GET CONTROL CHAIN HEAD (DETAIL DEF.)
IS CONTROL CH. HEAD EQ. TO ANY DEF. ON D.D. LIST?
NO
ADD NEW ENTRY TO D.D. LIST
YES
IS NEW CONTROL TYPE LESS?
YES
NO
REPLACE D.D. LIST ENTRY BY NEW ENTRY
25


50
ANY MORE FIELDS TO MODIFY?
YES
5
NO
GET LOC. OF CURRENT R.D. FROM .QCURD
GET REF. CODE OF CURRENT REC. → ARG-1 of .QTLNK
ANY ENTRIES IN DETAIL DEF. LIST?
NO
YES
SET UP ARG-2 TO .QTLNK
CALL .QTLNK
ANY ERRORS?
YES
99
55
NO
GET NEXT OF FIELD CHAIN
IS IT A RECORD DEF?
YES
.QUPDC
99
NO
IS MODIFY SWITCH SET?
NO
YES
RETURN
RESET THE MODIFY SWITCH
SET UP ARGUMENTS +0 .QPACK
.QPACK MOVE W.S. TO REC.
55

.QMNO

FUNCTION:

Subroutine .QMNO finds space in a page for the specified record. It writes the record type, record size, and line number in the page and stores the reference code and the record type in the communication control block. In the page header record, the space available is reduced by the record size; and the corresponding line number flag is set.

CALLING SEQUENCE:

CALL .QMNO(ARG-1, ARG-2)

where ARG-1 contains a base page number in bits 18-35.

ARG-2 is nonzero if the record is a CALC chain detail; otherwise, it is zero.

CALLED BY:

.QSTOR, .QTLNK

CALLS FOR:

.QBIC, .QPACK, .QLOCK

LIMITATIONS AND RESTRICTIONS:

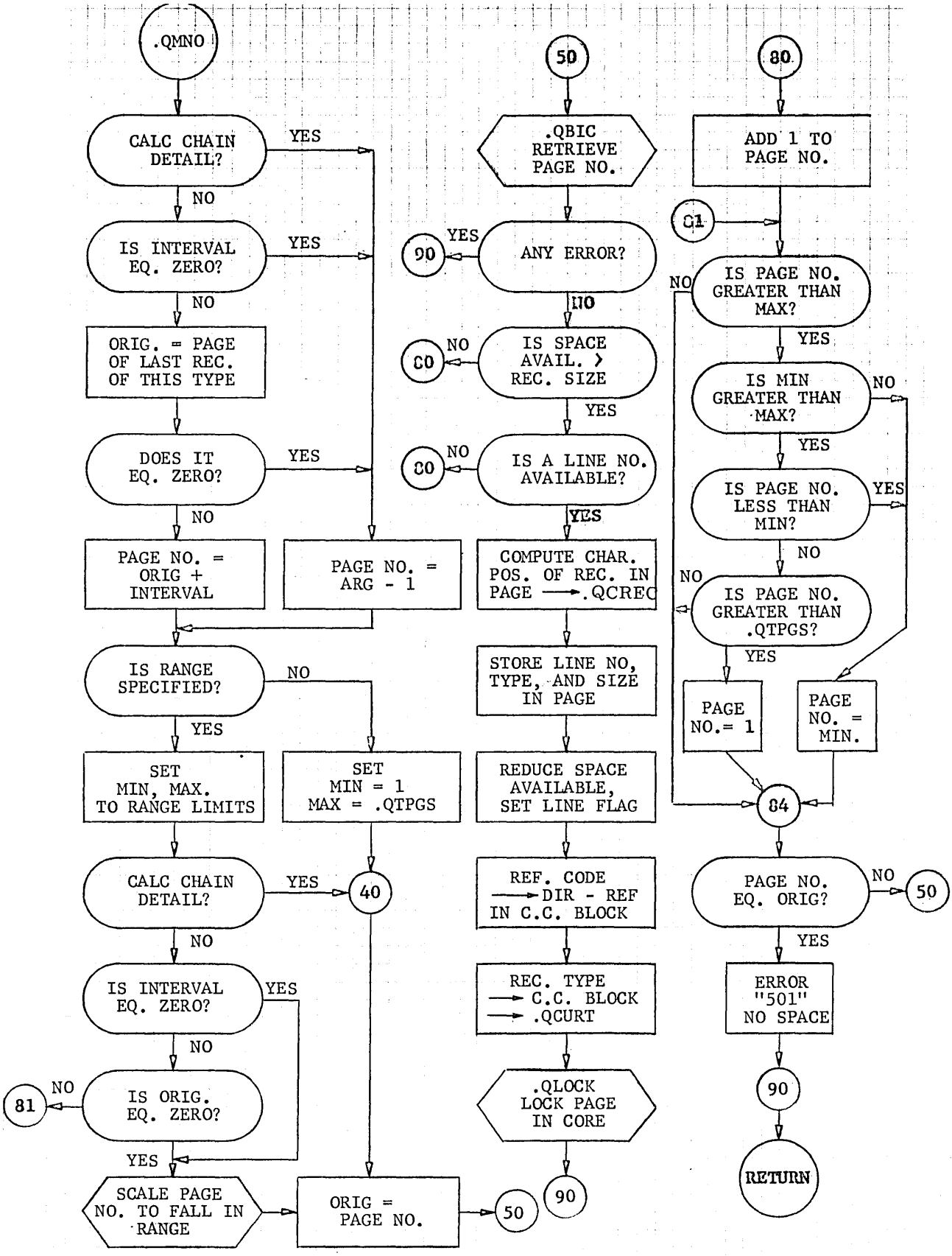
Index register 2 must contain the location of the record definition of the record being stored.

NUMBER OF WORDS:

307 octal

DETECTED ERRORS:

S01	Attempt to store a record when there is no space available within the range of pages specified for the record.


.QMNO
CALC CHAIN DETAIL?
YES
NO
IS INTERVAL EQ. ZERO?
YES
NO
ORIG. = PAGE OF LAST REC. OF THIS TYPE
DOES IT EQ. ZERO?
YES
NO
PAGE NO. = ORIG + INTERVAL
PAGE NO. = ARG - 1
IS RANGE SPECIFIED?
NO
YES
SET MIN, MAX. TO RANGE LIMITS
SET MIN = 1 MAX = .QTPGS
CALC CHAIN DETAIL?
YES
40
NO
IS INTERVAL EQ. ZERO?
YES
NO
81
NO
IS ORIG. EQ. ZERO?
YES
SCALE PAGE NO. TO FALL IN RANGE
ORIG = PAGE NO.
50
90
50
.QBIC RETRIEVE PAGE NO.
90
YES
ANY ERROR?
NO
80
NO
IS SPACE AVAIL. > REC. SIZE
YES
80
NO
IS A LINE NO. AVAILABLE?
YES
COMPUTE CHAR. POS. OF REC. IN PAGE → .QCREC
STORE LINE NO, TYPE, AND SIZE IN PAGE
REDUCE SPACE AVAILABLE, SET LINE FLAG
REF. CODE → DIR - REF IN C.C. BLOCK
REC. TYPE → C.C. BLOCK → .QCURT
.QLOCK LOCK PAGE IN CORE
90
50
80
ADD 1 TO PAGE NO.
81
IS PAGE NO. GREATER THAN MAX?
NO
YES
IS MIN GREATER THAN MAX?
NO
YES
IS PAGE NO. LESS THAN MIN?
YES
NO
NO
IS PAGE NO. GREATER THAN .QTPGS?
YES
PAGE NO.= 1
PAGE NO. = MIN.
84
PAGE NO. EQ. ORIG?
NO
50
YES
ERROR "501" NO SPACE
90
RETURN

.QMOVE

FUNCTION:

Subroutine .QMOVE unpacks data fields of the record last processed and moves all fields or selectively defined fields to working storage according to the field definition. If a selective field list is specified, only those fields indicated are moved to working storage.

CALLING SEQUENCE:

CALL .QMOVE(ARG-1, ARG-2)

where ARG-1 is zero or the location of the first word of a selective field list; if ARG-1 is zero, all fields of the record are moved to working storage.

ARG-2 contains a binary integer equal to the number of fields to be moved when a selective field list is specified. If ARG-1 is zero, ARG-2 is also zero.

CALLED BY:

Source program, .QHEAD, .QDLTE

CALLS FOR:

.QAUTH, .QUIT, .QPACK

NUMBER OF WORDS:

117 octal

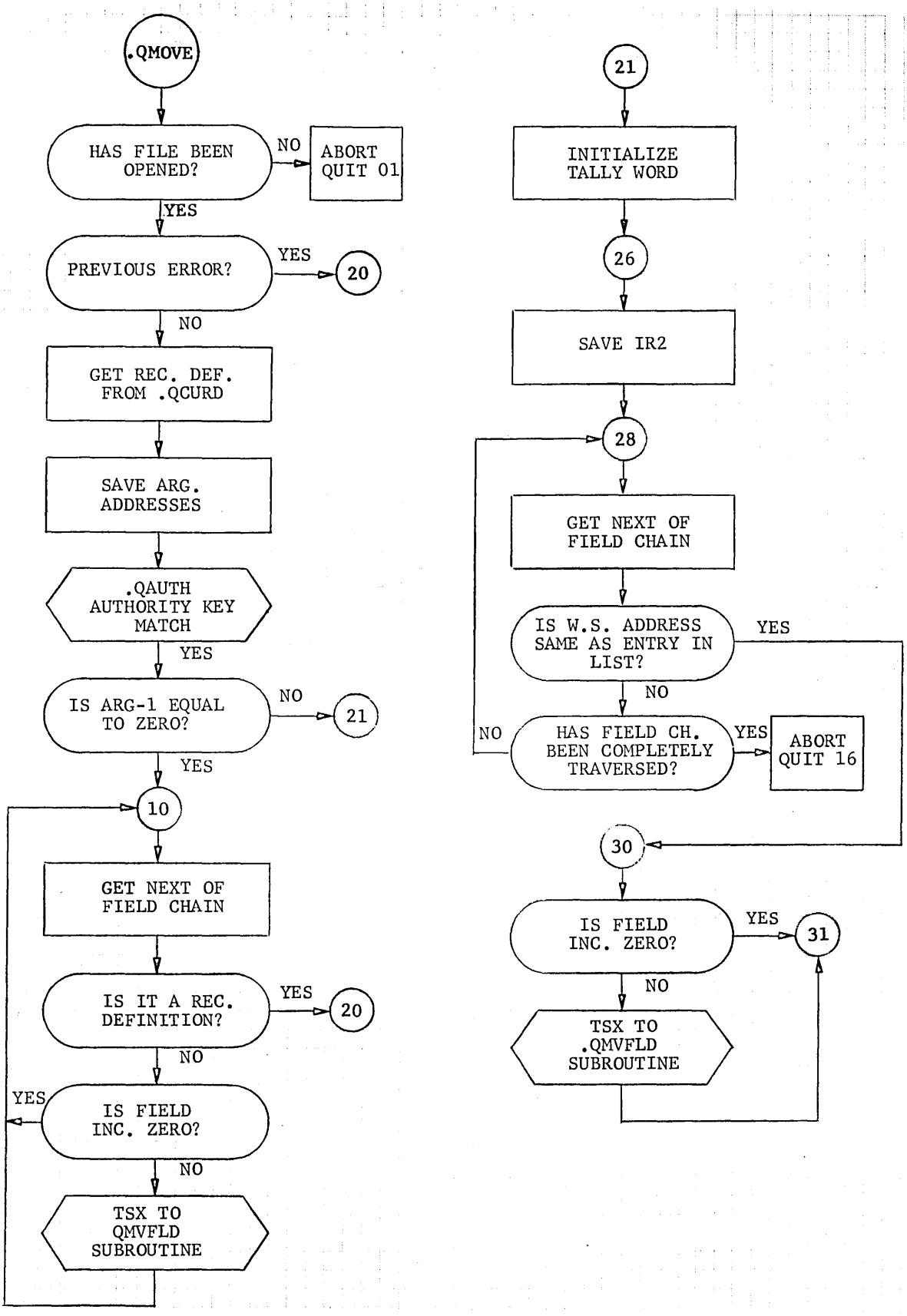
LIMITATIONS AND RESTRICTIONS:

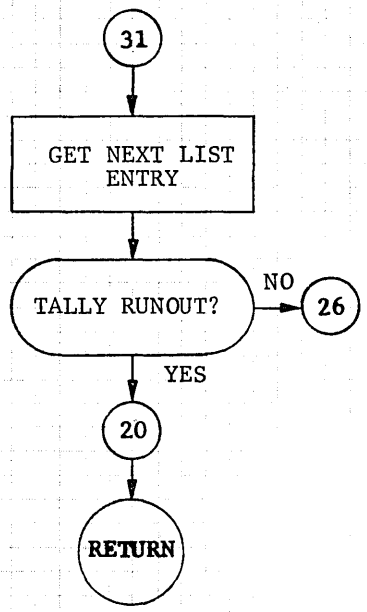
This routine checks the contents of ERROR-REFERENCE to determine whether any error occurred during or since the retrieval of the record involved. If the contents are nonzero, the function of the routine is aborted, and an immediate return to the instruction following the call is executed.

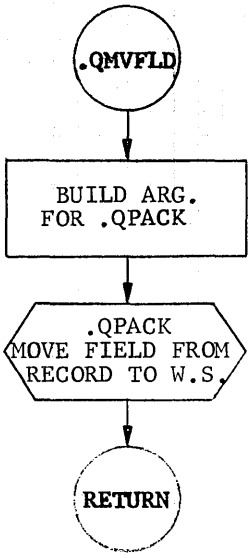
In addition, if an error occurs during a selective field movement, all additional fields specified in the list are moved to working storage before the return to the calling program.

DETECTED ERRORS:

QUIT (01)	Attempt to store or retrieve a record with the IDS file not opened.
QUIT (16)	Field to be modified or moved not defined for the current record.


.QMOVE
HAS FILE BEEN OPENED?
NO
ABORT QUIT 01
YES
PREVIOUS ERROR?
YES
20
NO
GET REC. DEF. FROM .QCURD
SAVE ARG. ADDRESSES
.QAUTH AUTHORITY KEY MATCH
YES
IS ARG-1 EQUAL TO ZERO?
NO
21
YES
10
GET NEXT OF FIELD CHAIN
IS IT A REC. DEFINITION?
YES
20
NO
YES
IS FIELD INC. ZERO?
NO
TSX TO QMVFLD SUBROUTINE
21
INITIALIZE TALLY WORD
26
SAVE IR2
28
GET NEXT OF FIELD CHAIN
IS W.S. ADDRESS SAME AS ENTRY IN LIST?
YES
NO
NO
HAS FIELD CH. BEEN COMPLETELY TRAVERSED?
YES
ABORT QUIT 16
30
IS FIELD INC. ZERO?
YES
31
NO
TSX TO .QMVFLD SUBROUTINE


31
GET NEXT LIST ENTRY
TALLY RUNOUT?
NO
26
YES
20
RETURN


.QMVFLD
BUILD ARG. FOR .QPACK
.QPACK MOVE FIELD FROM RECORD TO W.S.
RETURN

.QMRAC

FUNCTION:

Subroutine .QMRAC retrieves the data record specified.

CALLING SEQUENCE:

CALL .QMRAC(ARG-1)

where ARG-1 is the location of the 24-bit reference code.

CALLED BY:

.QASC, .QGETC, .QGETD, .QGETE, .QTLNK, .QTYPX, .QDLTE, .QFWD

CALLS FOR:

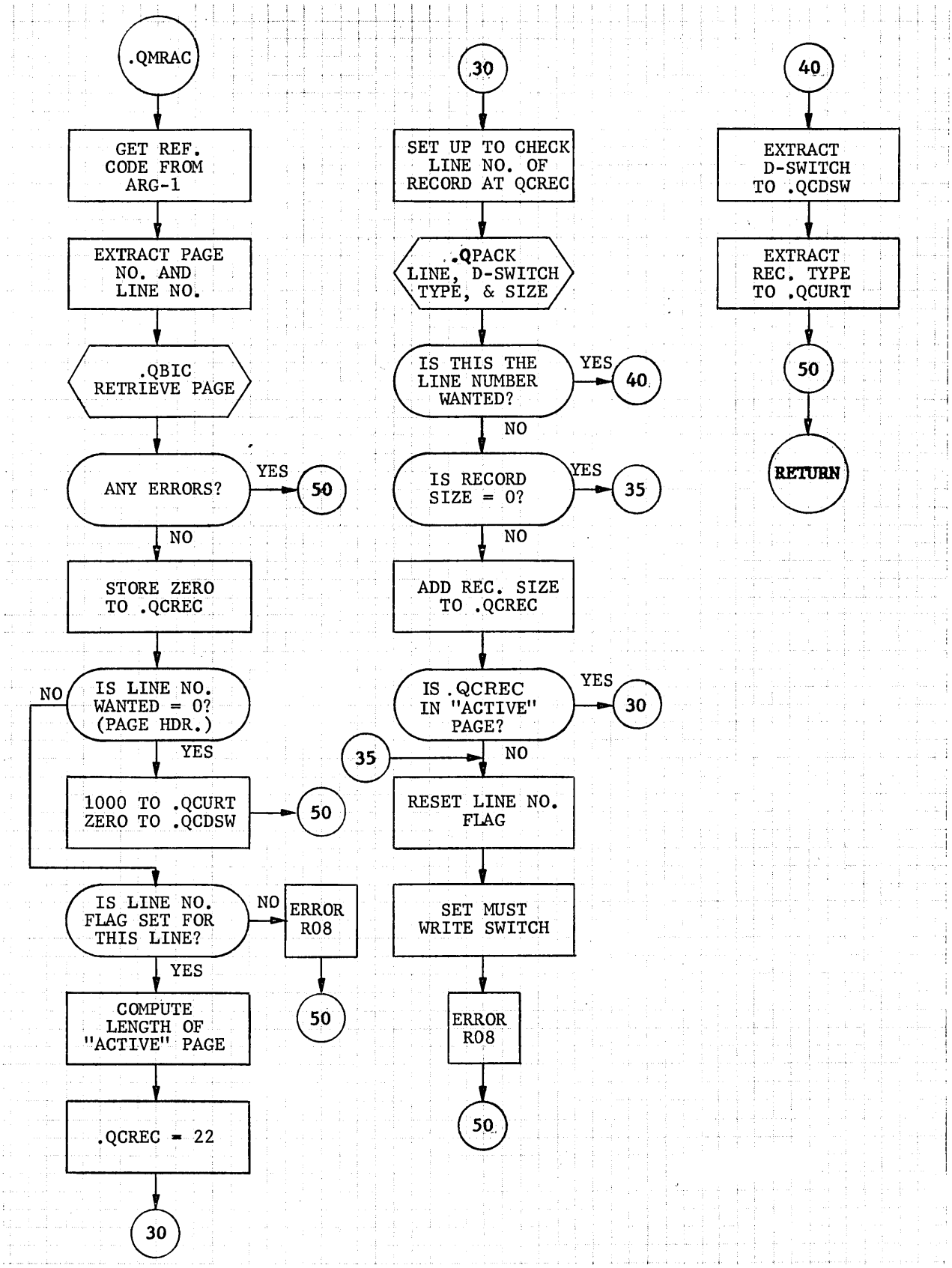
.QBIC, .QPACK

NUMBER OF WORDS:

146 octal

DETECTED ERRORS:

R08 Reference code of the record to be retrieved is not found within the specified page.


.QMRAC
GET REF. CODE FROM ARG-1
EXTRACT PAGE NO. AND LINE NO.
.QBIC RETRIEVE PAGE
ANY ERRORS?
YES
50
NO
STORE ZERO TO .QCREC
IS LINE NO. WANTED = 0? (PAGE HDR.)
NO
YES
1000 TO .QCURT ZERO TO .QCDSW
50
IS LINE NO. FLAG SET FOR THIS LINE?
NO
ERROR R08
50
YES
COMPUTE LENGTH OF "ACTIVE" PAGE
.QCREC = 22
30
30
SET UP TO CHECK LINE NO. OF RECORD AT QCREC
.QPACK LINE, D-SWITCH TYPE, & SIZE
IS THIS THE LINE NUMBER WANTED?
YES
40
NO
IS RECORD SIZE = 0?
YES
35
NO
ADD REC. SIZE TO .QCREC
IS .QCREC IN "ACTIVE" PAGE?
YES
30
35
NO
RESET LINE NO. FLAG
SET MUST WRITE SWITCH
ERROR R08
50
40
EXTRACT D-SWITCH TO .QCDSW
EXTRACT REC. TYPE TO .QCURT
50
RETURN

.QOPEN

FUNCTION:

Subroutine .QOPEN opens the IDS field, initializes various parameters, sets up the buffers in memory, and initializes the activity chain.

CALLING SEQUENCE:

CALL .QOPEN(ARG-1, ARG-3)

where ARG-1 is the location of the first word of the communication control block.

ARG-2 is the location of a word containing the authority key and the processing mode indicator.

ARG-3 is the location of the first record definition of the record-type chain.

CALLED BY:

Source program

CALLS FOR:

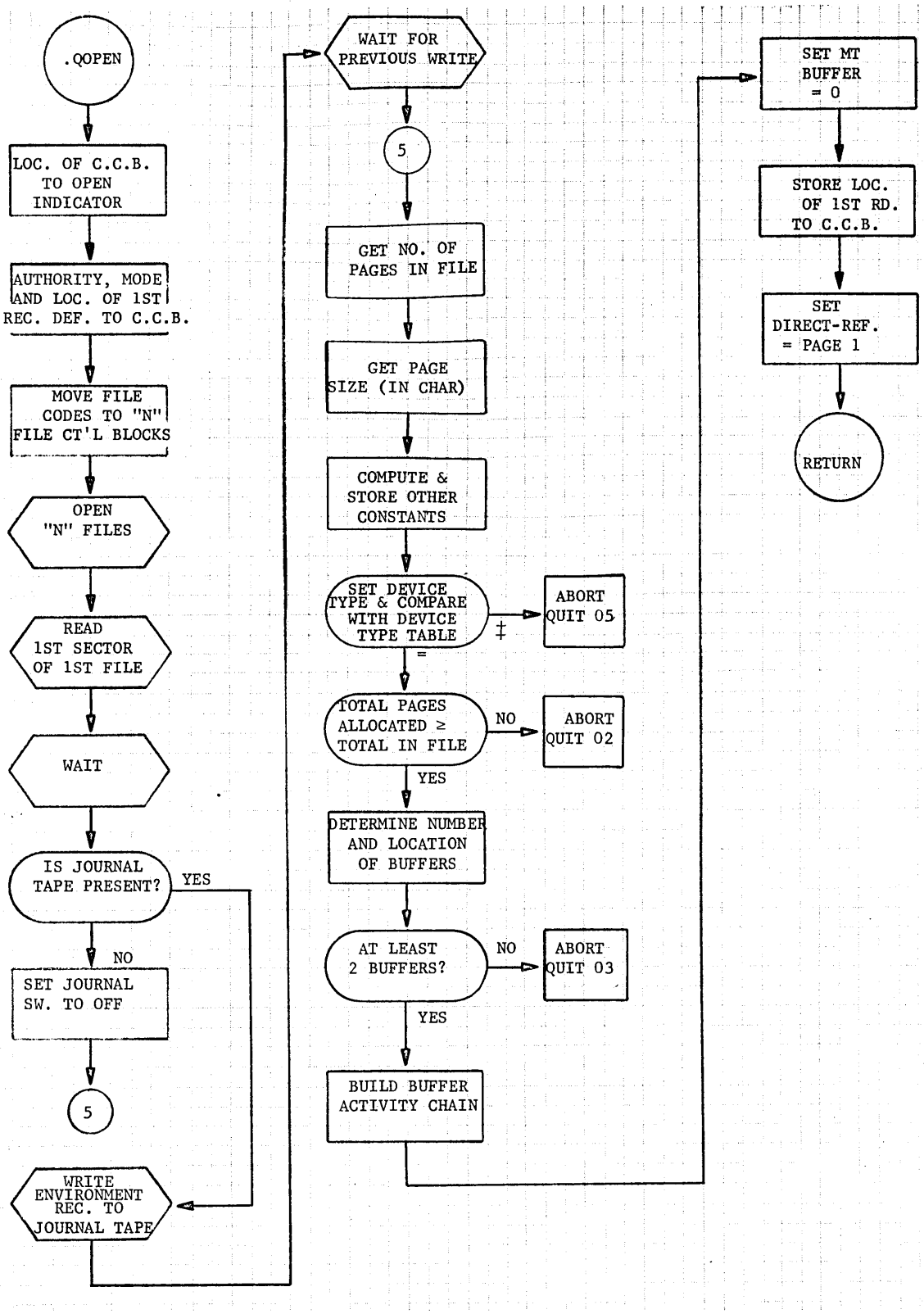
.GOPEN, .GREAD, .GWAIT, .QUIT

NUMBER OF WORDS:

460 octal

DETECTED ERRORS:

QUIT (02)	Capacity of the disc allocated via GECOS is not large enough for the number of pages specified by the environment record.
QUIT (03)	Insufficient memory allocated to provide at least two buffers.
QUIT (05)	Invalid device type allocation.


.QOPEN
LOC. OF C.C.B. TO OPEN INDICATOR
AUTHORITY, MODE AND LOC. OF 1ST REC. DEF. TO C.C.B.
MOVE FILE CODES TO "N" FILE CT'L BLOCKS
OPEN "N" FILES
READ 1ST SECTOR OF 1ST FILE
WAIT
IS JOURNAL TAPE PRESENT?
YES
NO
SET JOURNAL SW. TO OFF
5
WRITE ENVIRONMENT REC. TO JOURNAL TAPE
WAIT FOR PREVIOUS WRITE
5
GET NO. OF PAGES IN FILE
GET PAGE SIZE (IN CHAR)
COMPUTE & STORE OTHER CONSTANTS
SET DEVICE TYPE & COMPARE WITH DEVICE TYPE TABLE
‡
=
ABORT QUIT 05
TOTAL PAGES ALLOCATED ≥ TOTAL IN FILE
NO
ABORT QUIT 02
YES
DETERMINE NUMBER AND LOCATION OF BUFFERS
AT LEAST 2 BUFFERS?
NO
ABORT QUIT 03
YES
BUILD BUFFER ACTIVITY CHAIN
SET MT BUFFER = 0
STORE LOC. OF 1ST RD. TO C.C.B.
SET DIRECT-REF. = PAGE 1
RETURN

.QPACK

FUNCTION:

Subroutine .QPACK moves a field, character by character, from any source to any destination in core.

CALLING SEQUENCE:

CALL .QPACK(ARG-1, ARG-2, ARG-3)

where ARG-1 contains the source field's word location in 0-17 and the field's relative character position in 18-35.

ARG-2 contains the destination field's word location in 0-17 and the field's relative character position in 18-35.

ARG-3 contains the field size in characters; it is right justified.

CALLED BY:

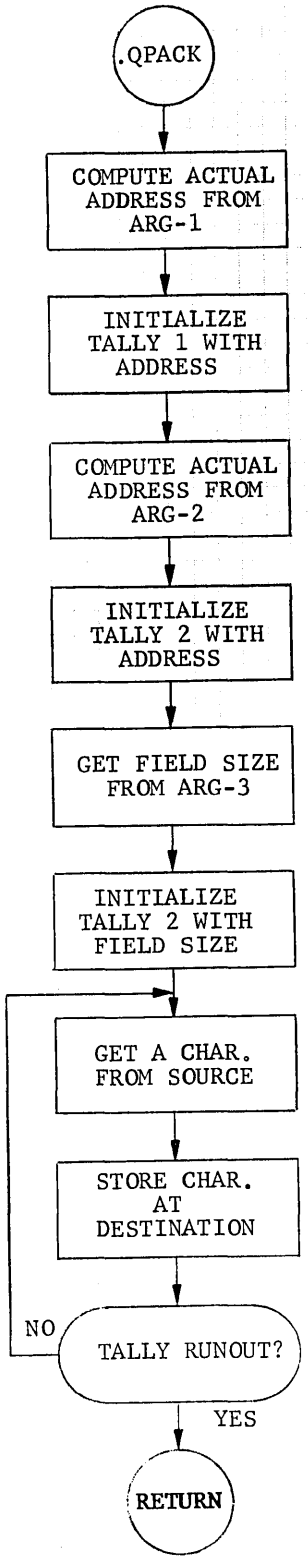
.QDLNK, .QFWD, .QGDET, .QHEAD, .QMNO, .QMOVE, .QMRAC, .QRUND, .QSTOR, .QTLNK, .QTYPX, .QUDCH, .QADJU, .QDLTE, .QMDFY

CALLS FOR:

Not applicable

NUMBER OF WORDS:

37 octal


.QPACK
COMPUTE ACTUAL ADDRESS FROM ARG-1
INITIALIZE TALLY 1 WITH ADDRESS
COMPUTE ACTUAL ADDRESS FROM ARG-2
INITIALIZE TALLY 2 WITH ADDRESS
GET FIELD SIZE FROM ARG-3
INITIALIZE TALLY 2 WITH FIELD SIZE
GET A CHAR. FROM SOURCE
STORE CHAR. AT DESTINATION
NO
TALLY RUNOUT?
YES
RETURN

.QRLS

FUNCTION:

Subroutine .QRLS unlocks a page in core by altering the entry in the activity chain corresponding to the buffer containing the page and by altering other entries. In this manner, the buffer containing the pages becomes the one of most recent activity. If the buffer was locked more than once, the subroutine increases the value on the activity chain entry by 1.

CALLING SEQUENCE:

CALL .QRLS(ARG-1)

where ARG-1 is a word containing the number of the buffer to be unlocked, in bits 0-17.

CALLED BY:

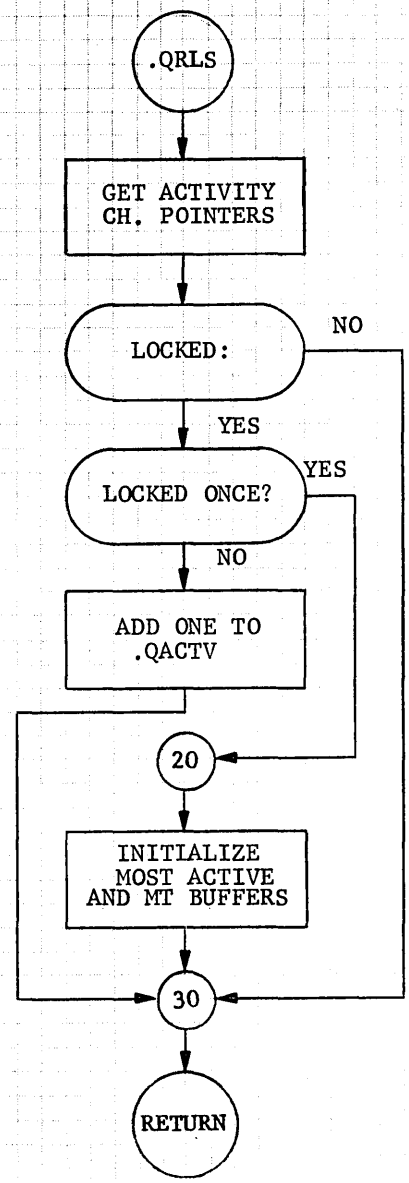
.QBIC, .QSTOR

CALLS FOR:

Not applicable

NUMBER OF WORDS:

35 octal


.QRLS
GET ACTIVITY CH. POINTERS
LOCKED:
NO
YES
LOCKED ONCE?
YES
NO
ADD ONE TO .QACTV
20
INITIALIZE MOST ACTIVE AND MT BUFFERS
30
RETURN

.QRUND

FUNCTION:

Subroutine .QRUND determines the reference code of the record just prior to the chain table CURRENT by going around the chain in a forward direction.

CALLING SEQUENCE:

```
CALL   .QRUND
```

CALLED BY:

.QDLNK, .QFWD, .QTLNK

CALLS FOR:

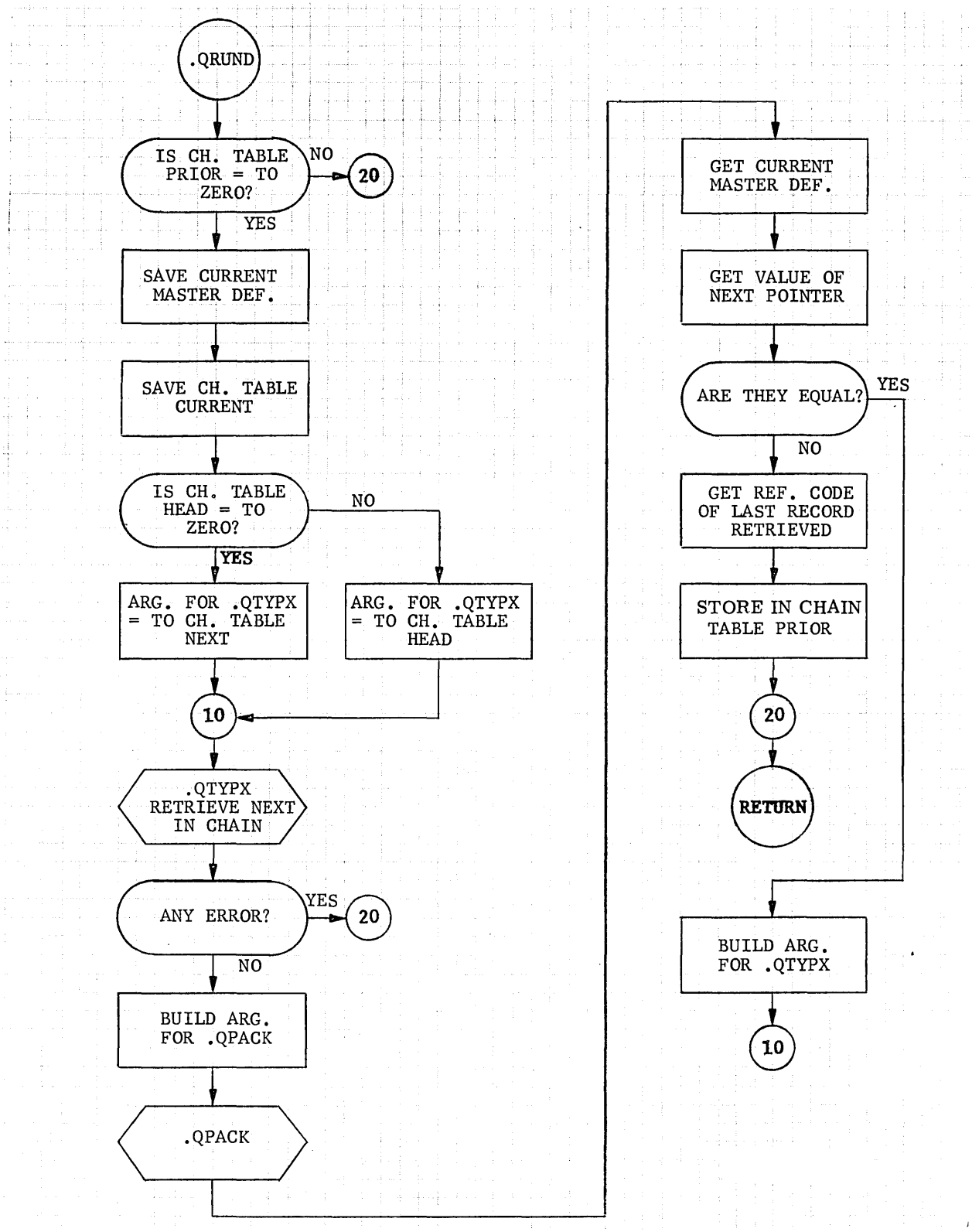
.QTYPX, .QPACK

LIMITATIONS AND RESTRICTIONS:

Index register 2 must contain the location of the master definition of the chain.

NUMBER OF WORDS:

65 octal


.QRUND
IS CH. TABLE PRIOR = TO ZERO?
NO
20
YES
SAVE CURRENT MASTER DEF.
SAVE CH. TABLE CURRENT
IS CH. TABLE HEAD = TO ZERO?
NO
YES
ARG. FOR .QTYPX = TO CH. TABLE NEXT
ARG. FOR .QTYPX = TO CH. TABLE HEAD
10
.QTYPX RETRIEVE NEXT IN CHAIN
ANY ERROR?
YES
20
NO
BUILD ARG. FOR .QPACK
.QPACK
GET CURRENT MASTER DEF.
GET VALUE OF NEXT POINTER
ARE THEY EQUAL?
YES
NO
GET REF. CODE OF LAST RECORD RETRIEVED
STORE IN CHAIN TABLE PRIOR
20
RETURN
BUILD ARG. FOR .QTYPX
10

.QSDSW

FUNCTION:

Subroutine .QSDSW sets the delete switch in the record last retrieved.

CALLING SEQUENCE:

```
CALL    .QSDSW
```

CALLED BY:

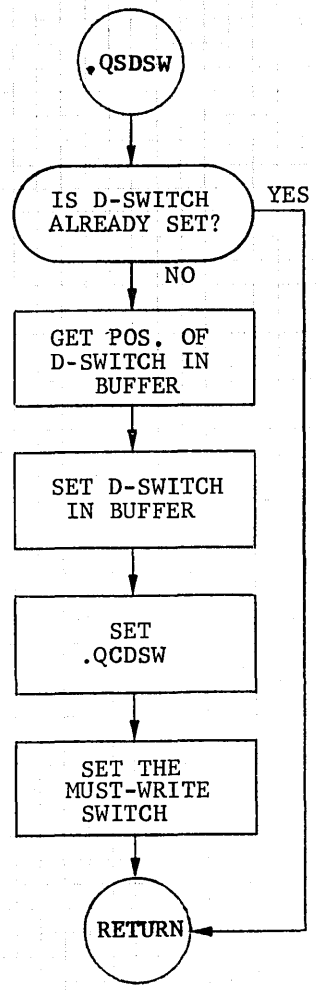
.QTLNK, .QDLTE

CALLS FOR:

Not applicable

NUMBER OF WORDS:

27 octal


.QSDSW
IS D-SWITCH ALREADY SET?
YES
NO
GET POS. OF D-SWITCH IN BUFFER
SET D-SWITCH IN BUFFER
SET .QCDSW
SET THE MUST-WRITE SWITCH
RETURN

.QSTOR

FUNCTION:

Subroutine .QSTOR places a given record into the IDS data file and links the record into all chains with which the record is associated.

CALLING SEQUENCE:

CALL .QSTOR(ARG-1)

where ARG-1 is the location of the record definition of the record to be stored.

CALLED BY:

Source program

CALLS FOR:

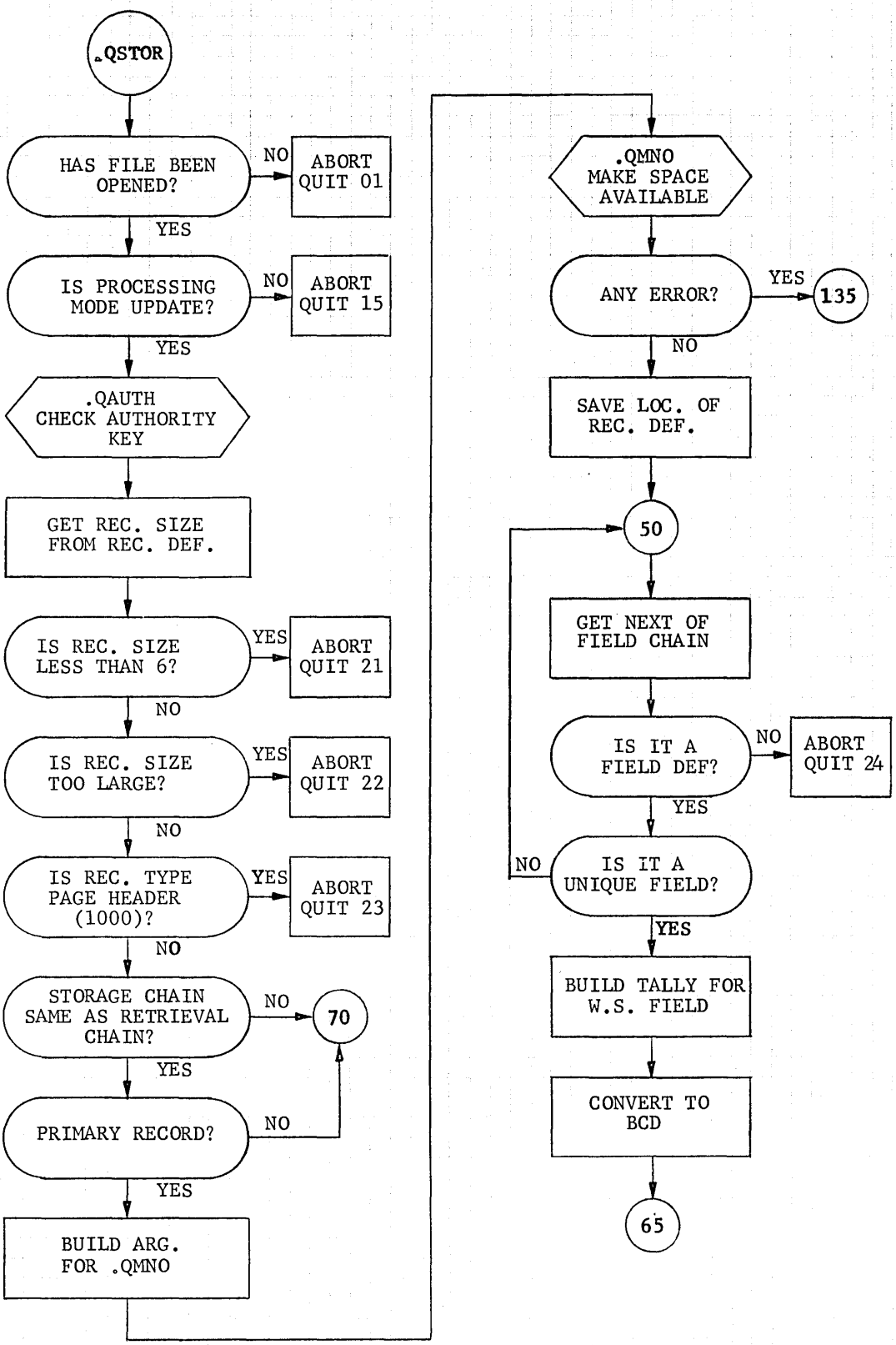
.QUIT, .QMNO, .QTLNK, .QADJU, .QRLS, .QPACK, .QUPDC

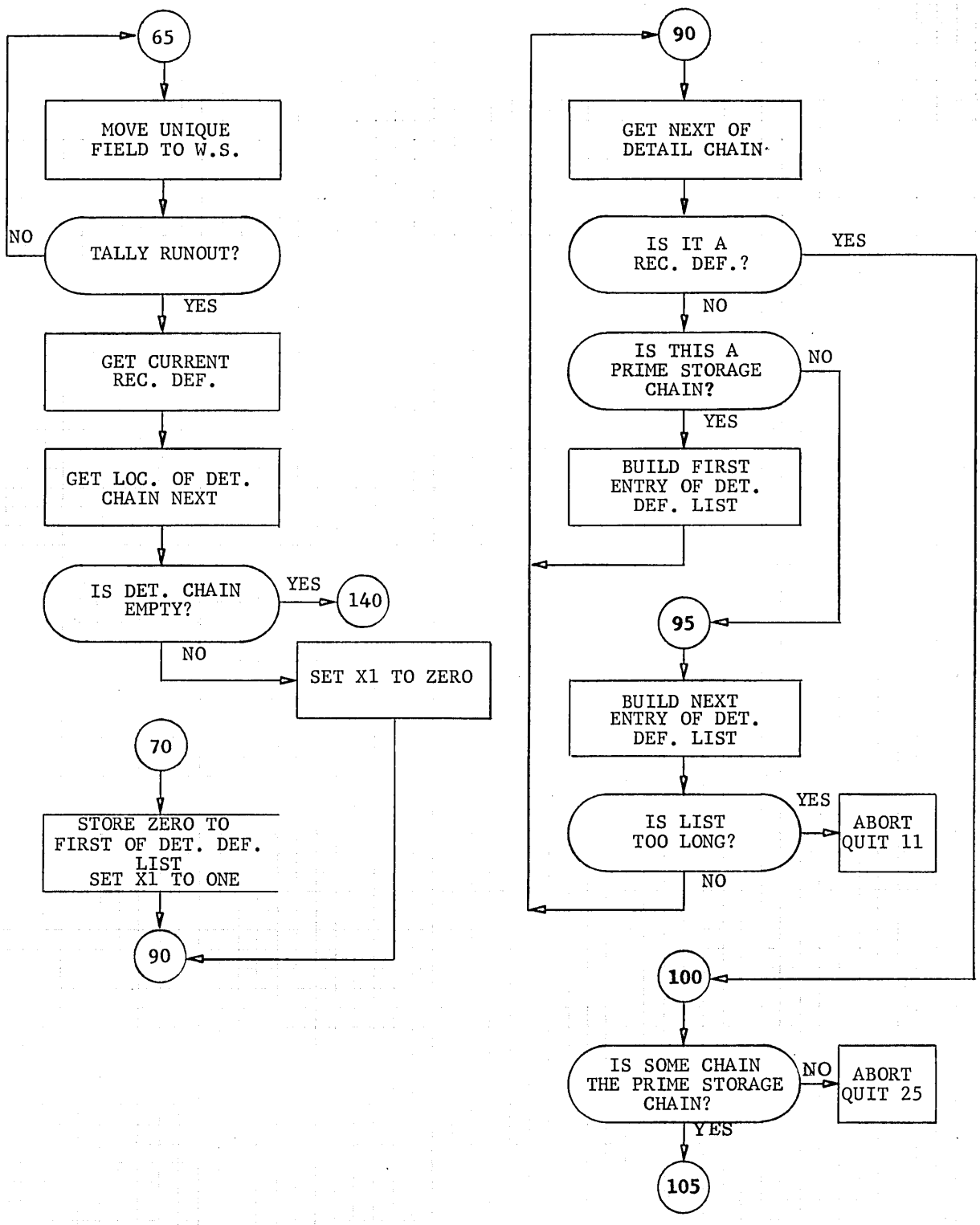
NUMBER OF WORDS:

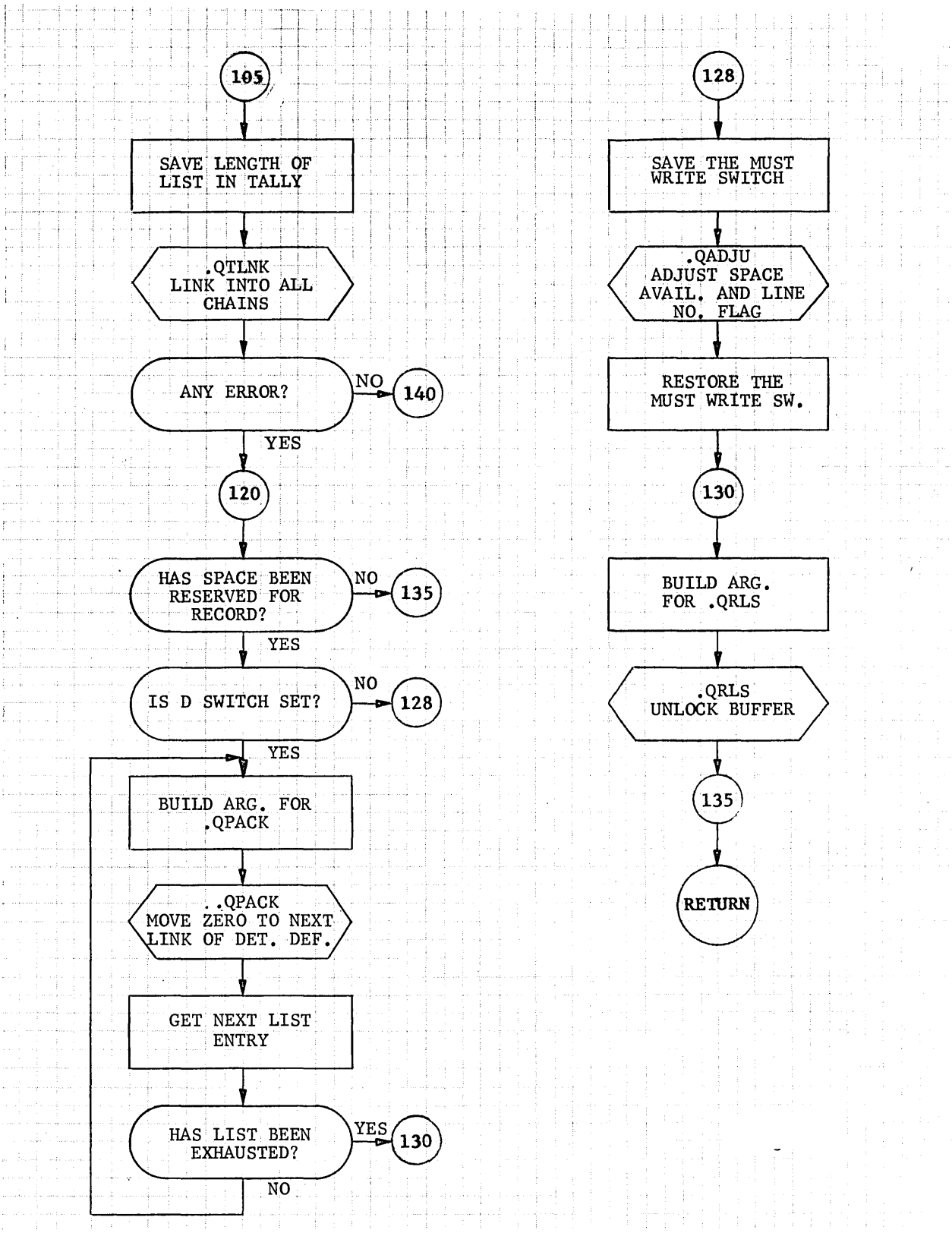
435 octal

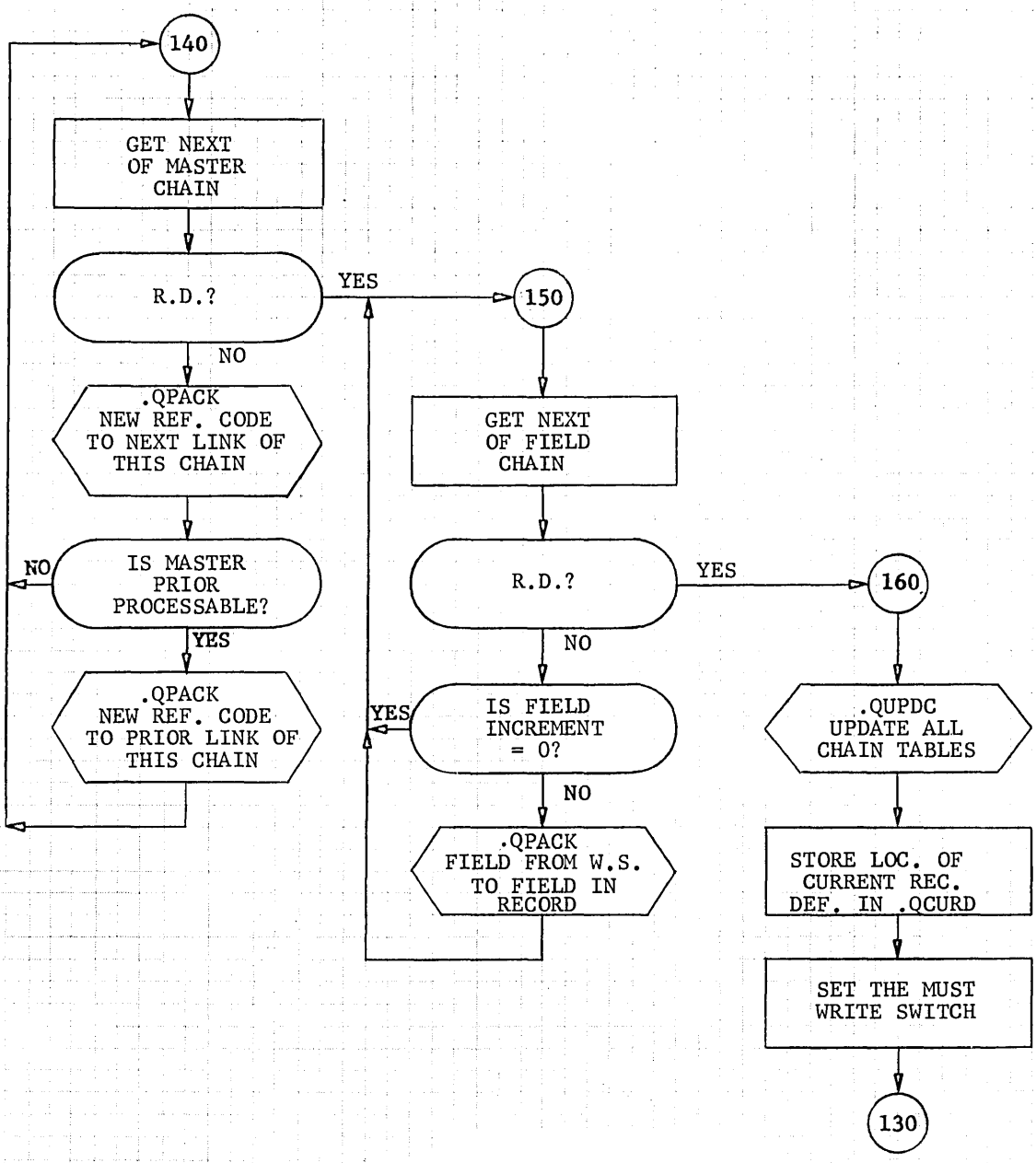
DETECTED ERRORS:

QUIT (01)	Attempt to store or retrieve a record with the IDS file not opened.
QUIT (11)	IDS subroutine table overflow. .QASC, .QDLTE, and .QSTOR must be reassembled.
QUIT (15)	Attempt to modify, delete, or store a record when the processing mode is not update.
QUIT (21)	Record definition of a record to be stored indicates that the record is less than 6 characters long.
QUIT (22)	Record definition of a record to be stored indicates that the record is greater than page size minus 22.
QUIT (23)	Attempt to store a page header record.
QUIT (24)	Attempt to store a primary record with no unique field defined.
QUIT (25)	Attempt to store a record which is a detail in one or more chains, when no storage chain is specified.


.QSTOR
HAS FILE BEEN OPENED?
NO
ABORT QUIT 01
YES
IS PROCESSING MODE UPDATE?
NO
ABORT QUIT 15
YES
.QAUTH CHECK AUTHORITY KEY
GET REC. SIZE FROM REC. DEF.
IS REC. SIZE LESS THAN 6?
YES
ABORT QUIT 21
NO
IS REC. SIZE TOO LARGE?
YES
ABORT QUIT 22
NO
IS REC. TYPE PAGE HEADER (1000)?
YES
ABORT QUIT 23
NO
STORAGE CHAIN SAME AS RETRIEVAL CHAIN?
NO
70
YES
PRIMARY RECORD?
NO
YES
BUILD ARG. FOR .QMNO
.QMNO MAKE SPACE AVAILABLE
ANY ERROR?
YES
135
NO
SAVE LOC. OF REC. DEF.
50
GET NEXT OF FIELD CHAIN
IS IT A FIELD DEF?
NO
ABORT QUIT 24
YES
NO
IS IT A UNIQUE FIELD?
YES
BUILD TALLY FOR W.S. FIELD
CONVERT TO BCD
65


65
MOVE UNIQUE FIELD TO W.S.
NO
TALLY RUNOUT?
YES
GET CURRENT REC. DEF.
GET LOC. OF DET. CHAIN NEXT
IS DET. CHAIN EMPTY?
YES
140
NO
SET X1 TO ZERO
70
STORE ZERO TO FIRST OF DET. DEF. LIST
SET X1 TO ONE
90
90
GET NEXT OF DETAIL CHAIN
IS IT A REC. DEF.?
YES
NO
IS THIS A PRIME STORAGE CHAIN?
NO
YES
BUILD FIRST ENTRY OF DET. DEF. LIST
95
BUILD NEXT ENTRY OF DET. DEF. LIST
IS LIST TOO LONG?
YES
ABORT QUIT 11
NO
100
IS SOME CHAIN THE PRIME STORAGE CHAIN?
NO
ABORT QUIT 25
YES
105


105
SAVE LENGTH OF LIST IN TALLY
.QTLNK LINK INTO ALL CHAINS
ANY ERROR?
NO
140
YES
120
HAS SPACE BEEN RESERVED FOR RECORD?
NO
135
YES
IS D SWITCH SET?
NO
128
YES
BUILD ARG. FOR .QPACK
. .QPACK MOVE ZERO TO NEXT LINK OF DET. DEF.
GET NEXT LIST ENTRY
HAS LIST BEEN EXHAUSTED?
YES
130
NO
128
SAVE THE MUST WRITE SWITCH
.QADJU ADJUST SPACE AVAIL. AND LINE NO. FLAG
RESTORE THE MUST WRITE SW.
130
BUILD ARG. FOR .QRLS
.QRLS UNLOCK BUFFER
135
RETURN


140
GET NEXT OF MASTER CHAIN
R.D.?
YES
150
NO
.QPACK NEW REF. CODE TO NEXT LINK OF THIS CHAIN
GET NEXT OF FIELD CHAIN
IS MASTER PRIOR PROCESSABLE?
NO
YES
R.D.?
YES
160
NO
.QPACK NEW REF. CODE TO PRIOR LINK OF THIS CHAIN
YES
IS FIELD INCREMENT = 0?
.QUPDC UPDATE ALL CHAIN TABLES
NO
.QPACK FIELD FROM W.S. TO FIELD IN RECORD
STORE LOC. OF CURRENT REC. DEF. IN .QCURD
SET THE MUST WRITE SWITCH
130

.QSYN

FUNCTION:

Subroutine .QSYN exchanges the contents of the working storage area corresponding to a synonym field with that of the equivalent match key field.

CALLING SEQUENCE:

CALL .QSYN

CALLED BY:

.QASC, .QTLNK

CALLS FOR:

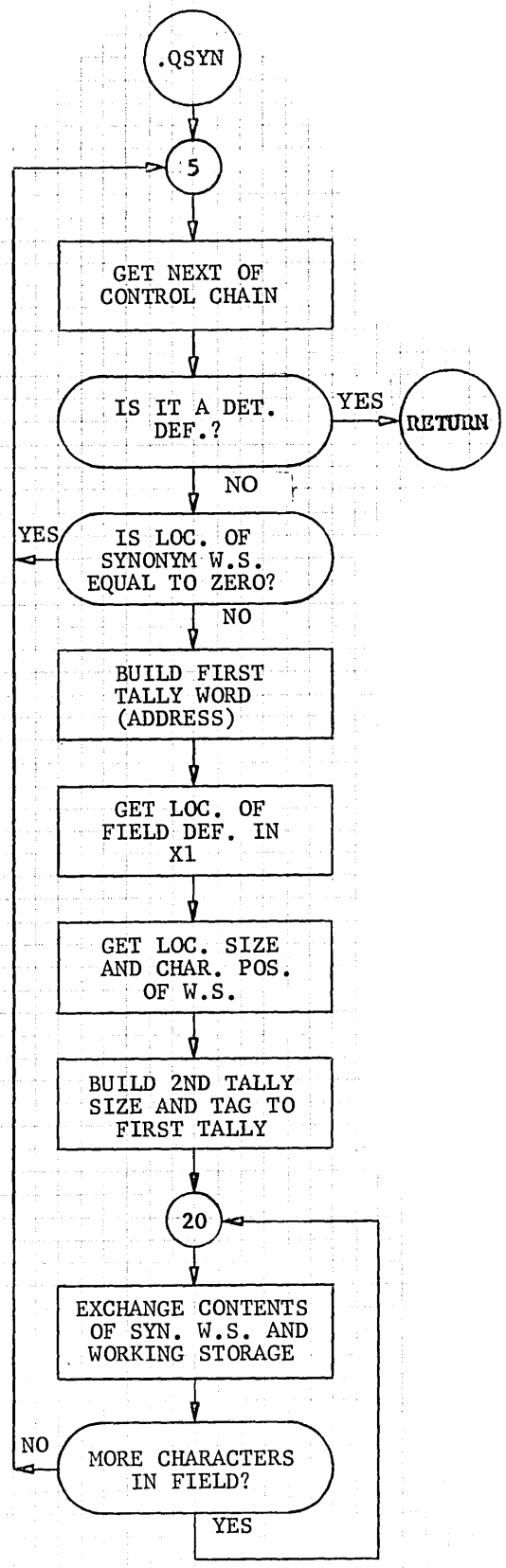
Not applicable

LIMITATIONS AND RESTRICTIONS:

Index register 2 must contain the location of the detail definition.

NUMBER OF WORDS:

35 octal


.QSYN
5
GET NEXT OF CONTROL CHAIN
IS IT A DET. DEF.?
YES
RETURN
NO
YES
IS LOC. OF SYNONYM W.S. EQUAL TO ZERO?
NO
BUILD FIRST TALLY WORD (ADDRESS)
GET LOC. OF FIELD DEF. IN X1
GET LOC. SIZE AND CHAR. POS. OF W.S.
BUILD 2ND TALLY SIZE AND TAG TO FIRST TALLY
20
EXCHANGE CONTENTS OF SYN. W.S. AND WORKING STORAGE
NO
MORE CHARACTERS IN FIELD?
YES

.QTLNK

FUNCTION:

Subroutine .QTLNK establishes linkages in all chains in which the specified record is a detail. If ARG-1 is zero, the record will be assigned storage space in some page.

CALLING SEQUENCE:

CALL .QTLNK(ARG-1, ARG-2)

where ARG-1 is zero or is the location of a word containing the reference code of the record to be linked.

ARG-2 is an indirect word in which the address field is the location of the first word of a list of entries describing detail definitions, the tally field contains the length of the list, and the tag field is zero.

CALLED BY:

.QSTOR, .QMDFY

CALLS FOR:

.QDLNK, .QCALC, .QSYN, .QASC, .QMRAC, .QUIT, .QUDCH, .QSDSW, .QTYPX, .QFWD, .QGDET, .QPACK, .QMNO, .QRUND

LIMITATIONS AND RESTRICTIONS:

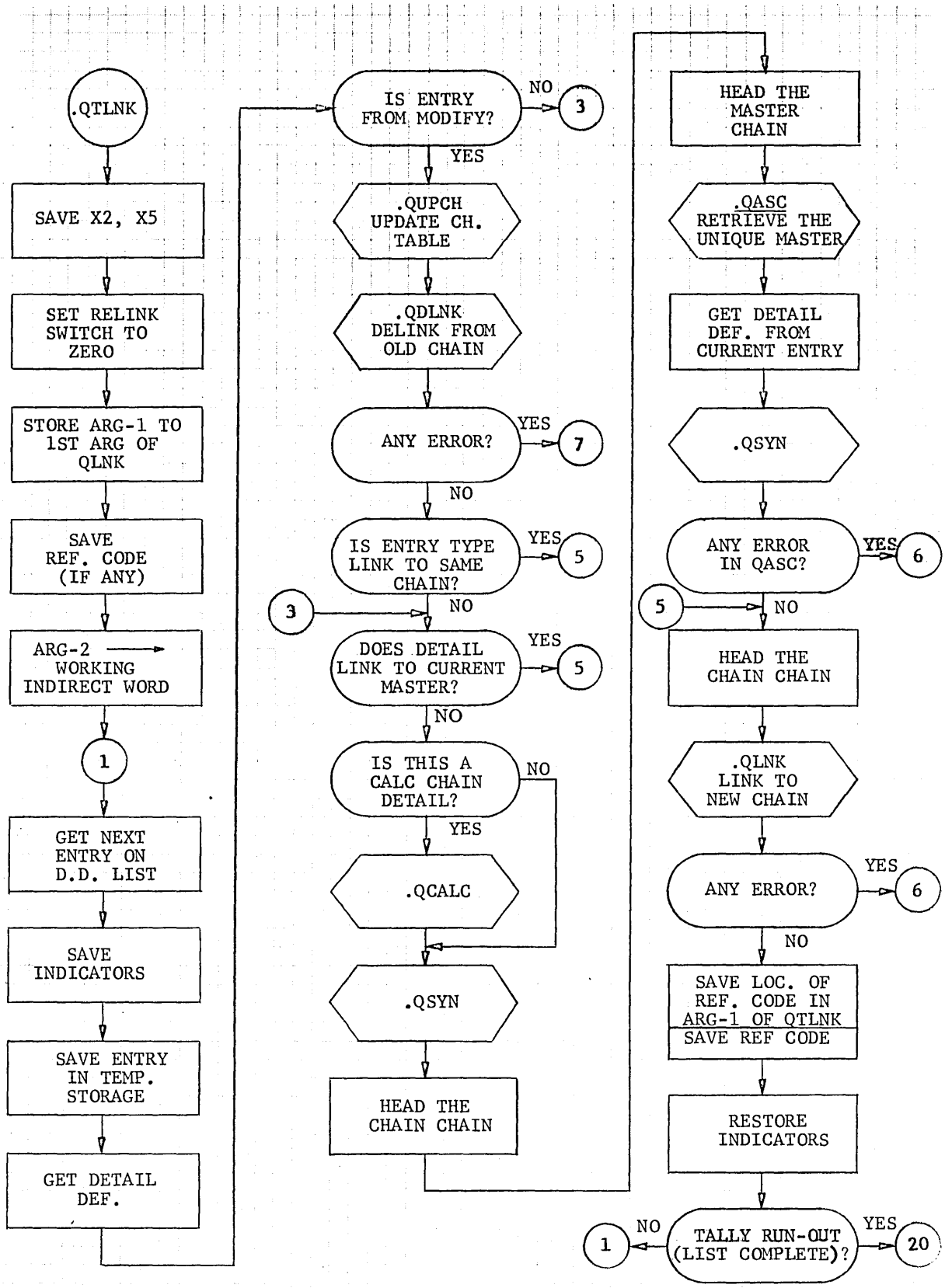
Index register 2 must contain the location of the record definition of the record to be linked.

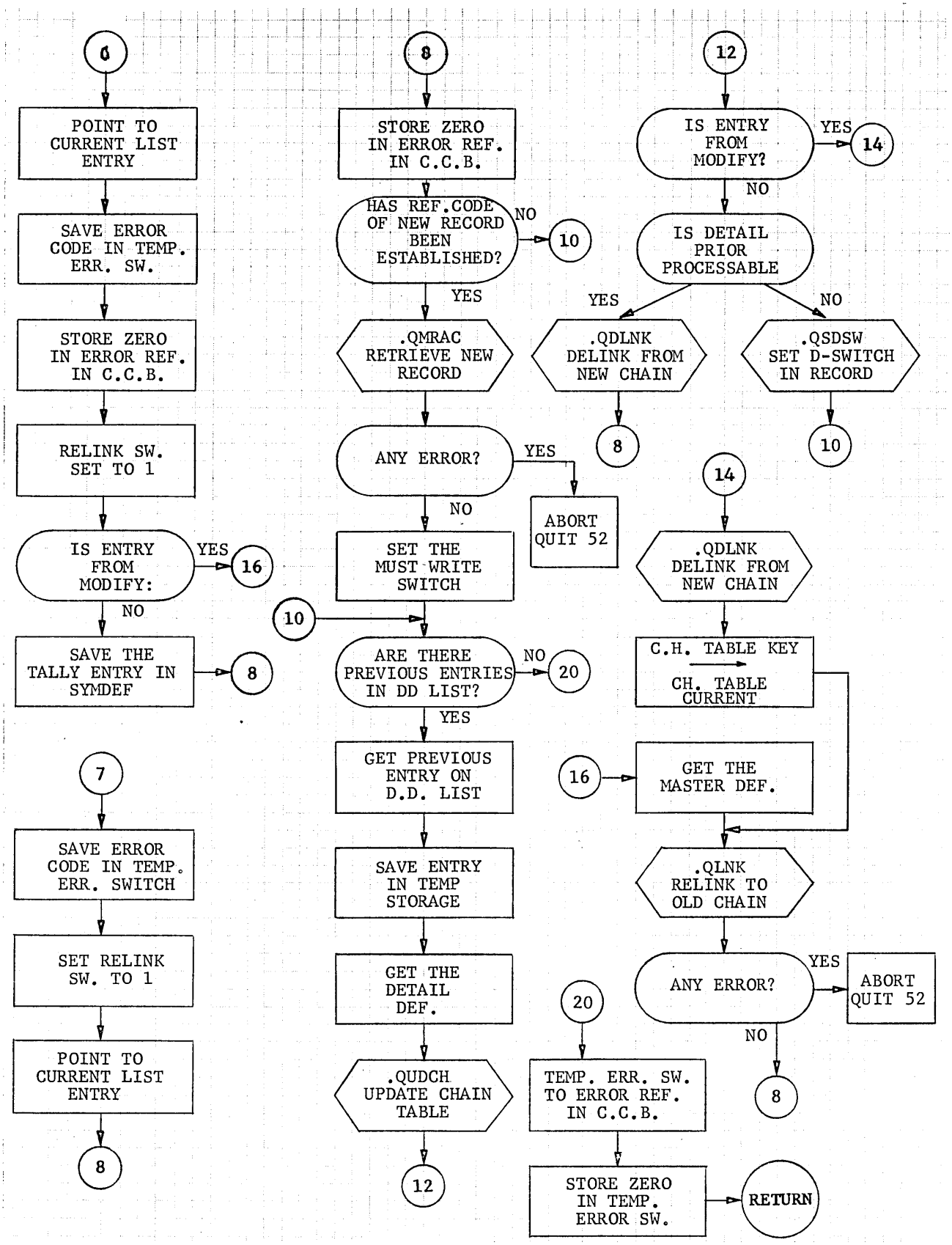
NUMBER OF WORDS:

746 octal

DETECTED ERRORS:

QUIT (52) Linking of a record cannot be completed. Either the record cannot be retrieved from the disc, or, for some other reason, the chain-pointer has been lost or is invalid.


.QTLNK
SAVE X2, X5
SET RELINK SWITCH TO ZERO
STORE ARG-1 TO 1ST ARG OF QLNK
SAVE REF. CODE (IF ANY)
ARG-2 → WORKING INDIRECT WORD
1
GET NEXT ENTRY ON D.D. LIST
SAVE INDICATORS
SAVE ENTRY IN TEMP. STORAGE
GET DETAIL DEF.
IS ENTRY FROM MODIFY?
NO
3
YES
.QUPCH UPDATE CH. TABLE
.QDLNK DELINK FROM OLD CHAIN
ANY ERROR?
YES
7
NO
IS ENTRY TYPE LINK TO SAME CHAIN?
YES
5
3
NO
DOES DETAIL LINK TO CURRENT MASTER?
YES
5
NO
IS THIS A CALC CHAIN DETAIL?
NO
YES
.QCALC
.QSYN
HEAD THE CHAIN CHAIN
HEAD THE MASTER CHAIN
.QASC RETRIEVE THE UNIQUE MASTER
GET DETAIL DEF. FROM CURRENT ENTRY
.QSYN
ANY ERROR IN QASC?
YES
6
5
NO
HEAD THE CHAIN CHAIN
.QLNK LINK TO NEW CHAIN
ANY ERROR?
YES
6
NO
SAVE LOC. OF REF. CODE IN ARG-1 OF QTLNK
SAVE REF CODE
RESTORE INDICATORS
TALLY RUN-OUT (LIST COMPLETE)?
NO
1
YES
20


6
POINT TO CURRENT LIST ENTRY
SAVE ERROR CODE IN TEMP. ERR. SW.
STORE ZERO IN ERROR REF. IN C.C.B.
RELINK SW. SET TO 1
IS ENTRY FROM MODIFY:
YES
16
NO
SAVE THE TALLY ENTRY IN SYMDEF
8
7
SAVE ERROR CODE IN TEMP. ERR. SWITCH
SET RELINK SW. TO 1
POINT TO CURRENT LIST ENTRY
8
8
STORE ZERO IN ERROR REF. IN C.C.B.
HAS REF.CODE OF NEW RECORD BEEN ESTABLISHED?
NO
10
YES
.QMRAC RETRIEVE NEW RECORD
ANY ERROR?
YES
ABORT QUIT 52
NO
SET THE MUST WRITE SWITCH
10
ARE THERE PREVIOUS ENTRIES IN DD LIST?
NO
20
YES
GET PREVIOUS ENTRY ON D.D. LIST
SAVE ENTRY IN TEMP STORAGE
GET THE DETAIL DEF.
.QUDCH UPDATE CHAIN TABLE
12
12
IS ENTRY FROM MODIFY?
YES
14
NO
IS DETAIL PRIOR PROCESSABLE
YES
NO
.QDLNK DELINK FROM NEW CHAIN
8
.QSDSW SET D-SWITCH IN RECORD
10
14
.QDLNK DELINK FROM NEW CHAIN
C.H. TABLE KEY → CH. TABLE CURRENT
16
GET THE MASTER DEF.
.QLNK RELINK TO OLD CHAIN
ANY ERROR?
YES
ABORT QUIT 52
NO
8
20
TEMP. ERR. SW. TO ERROR REF. IN C.C.B.
STORE ZERO IN TEMP. ERROR SW.
RETURN

.QLNK

FUNCTION:

Subroutine .QLNK links a detail into the specified chain. If necessary, a reference code will be established for the record.

CALLING SEQUENCE:

```
TSX1    .QLNK
ARG     ARG-1
ARG     ARG-2
```

where ARG-1 is zero or is the location of a word containing the reference code of the record to be linked.

ARG-2 contains the location of the detail definition in 0-17.

CALLED BY:

.QTLNK

CALLS FOR:

.QUIT, .QTYPX, .QGDET, .QRUND, .QPACK, .QUDCH, .QFWD, .QMNO

LIMITATIONS AND RESTRICTIONS:

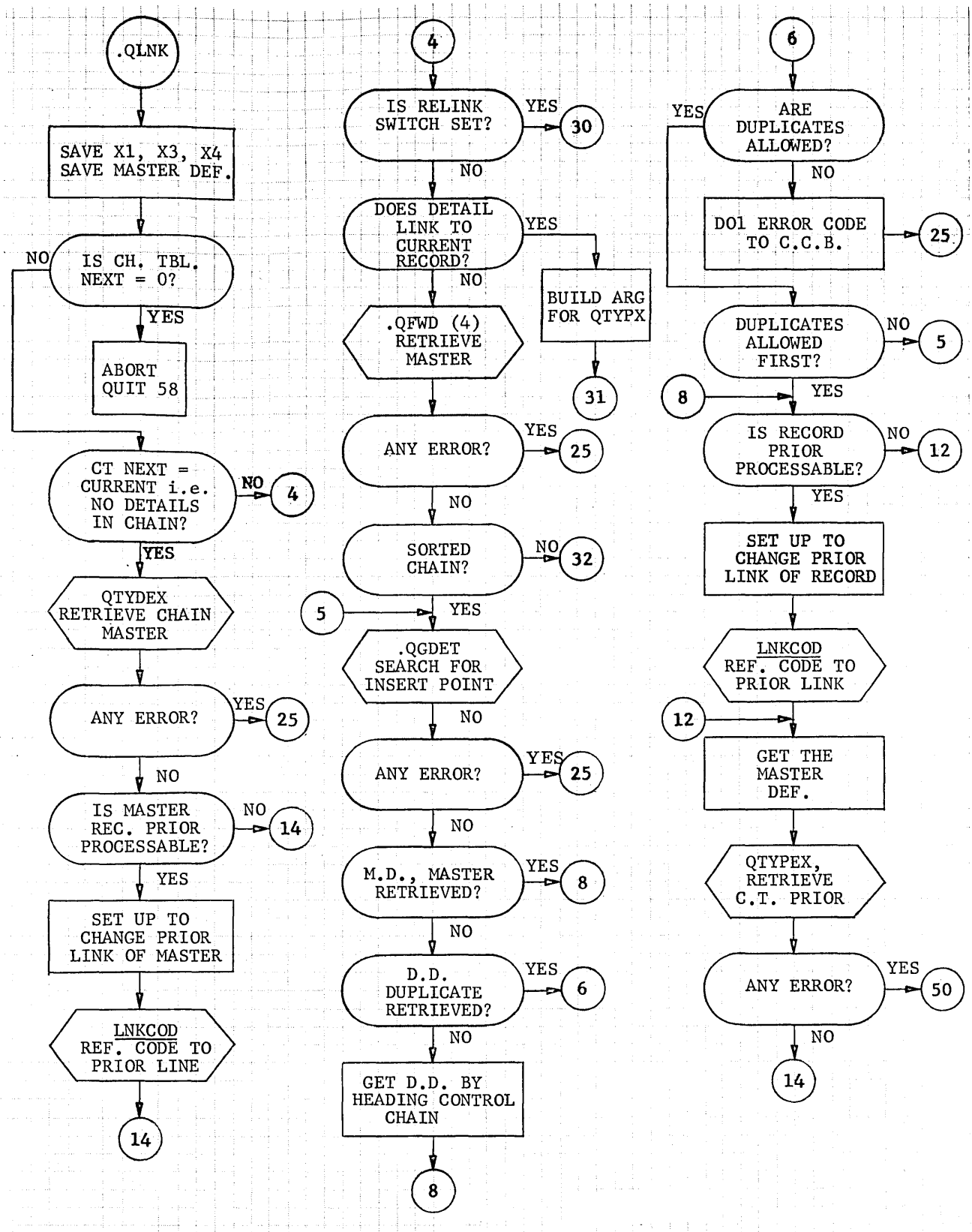
Index register 2 must contain the location of the master definition of the chain into which the record will be linked.

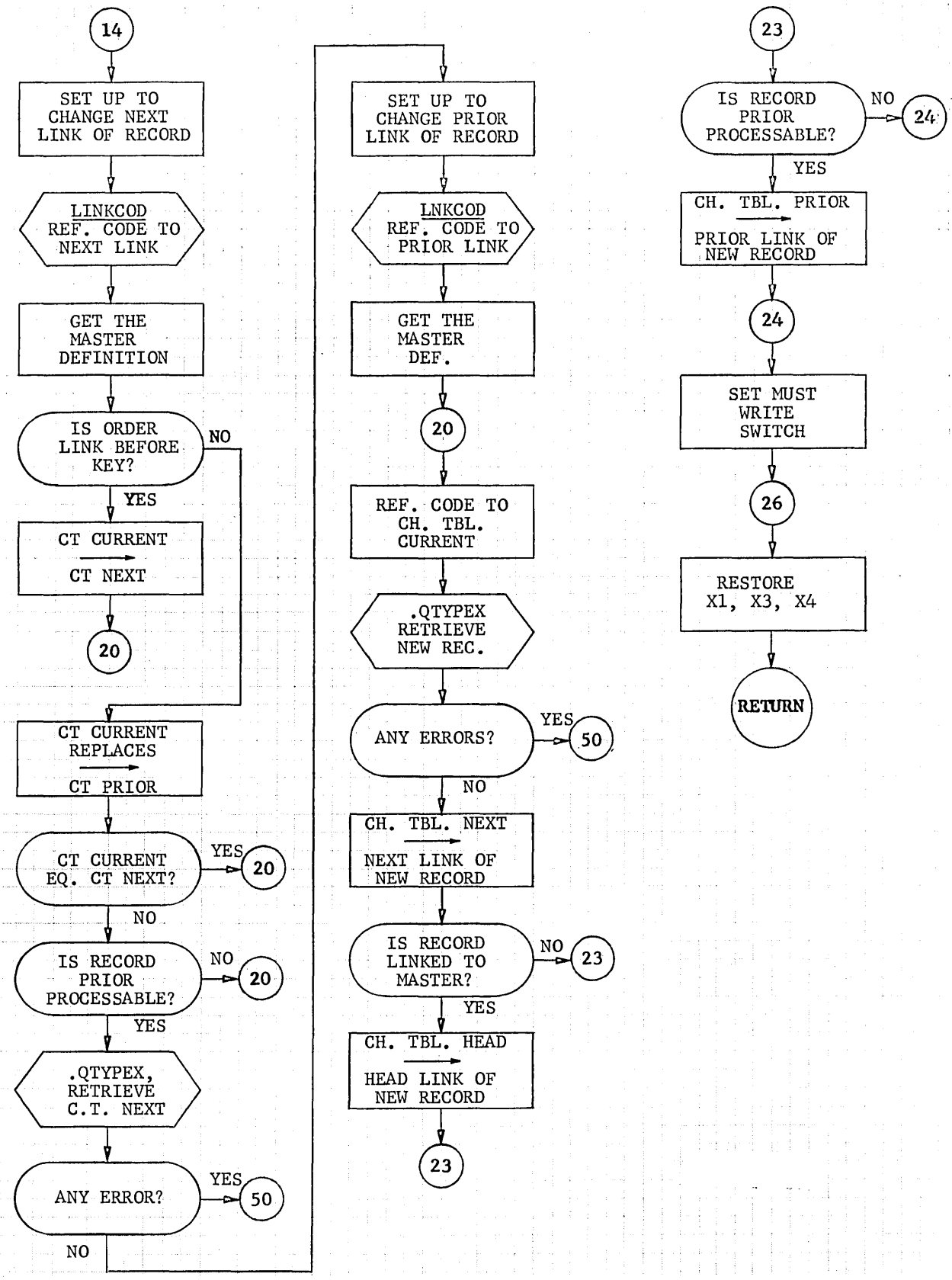
NUMBER OF WORDS:

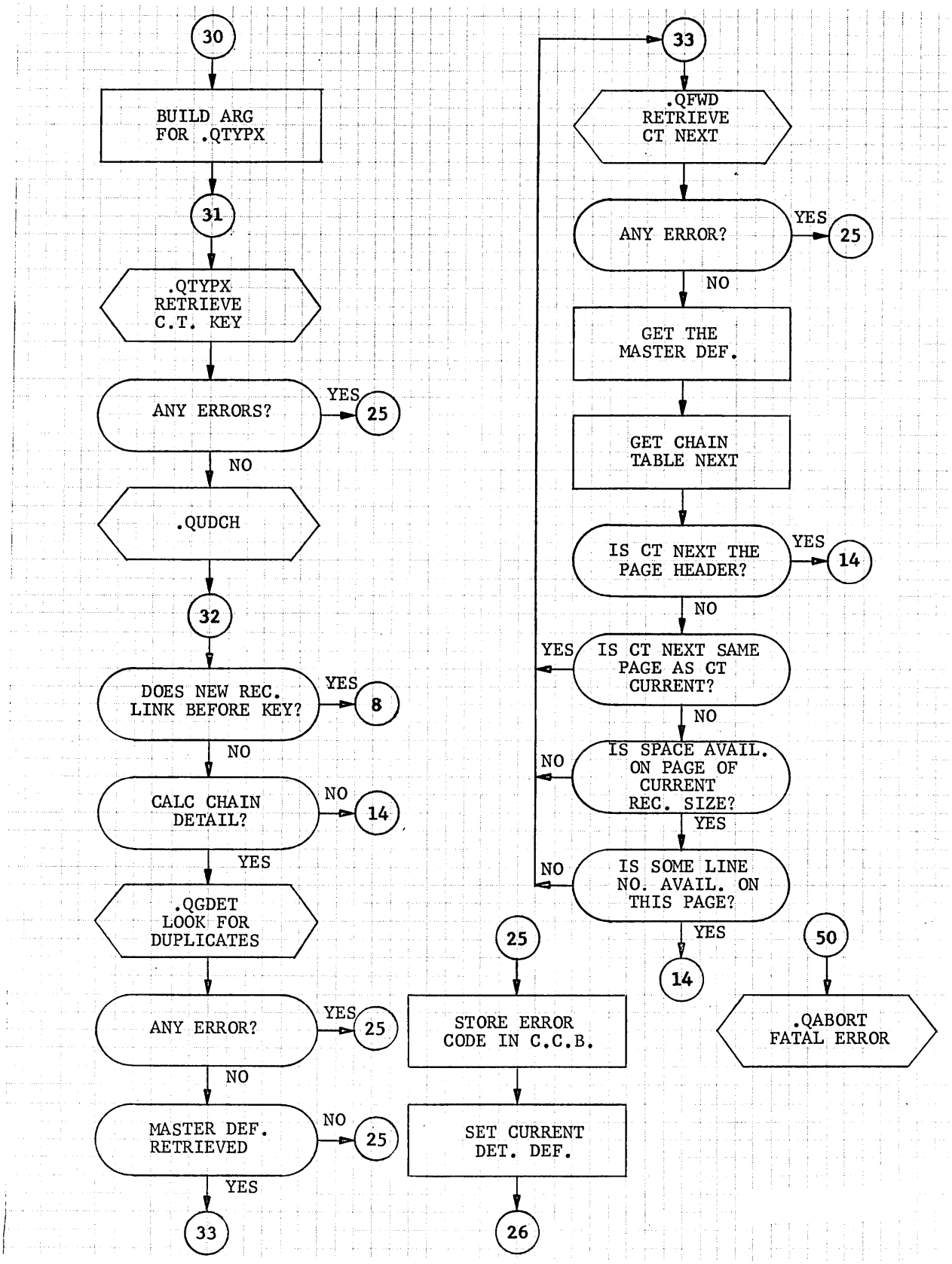
Included in .QTLNK total.

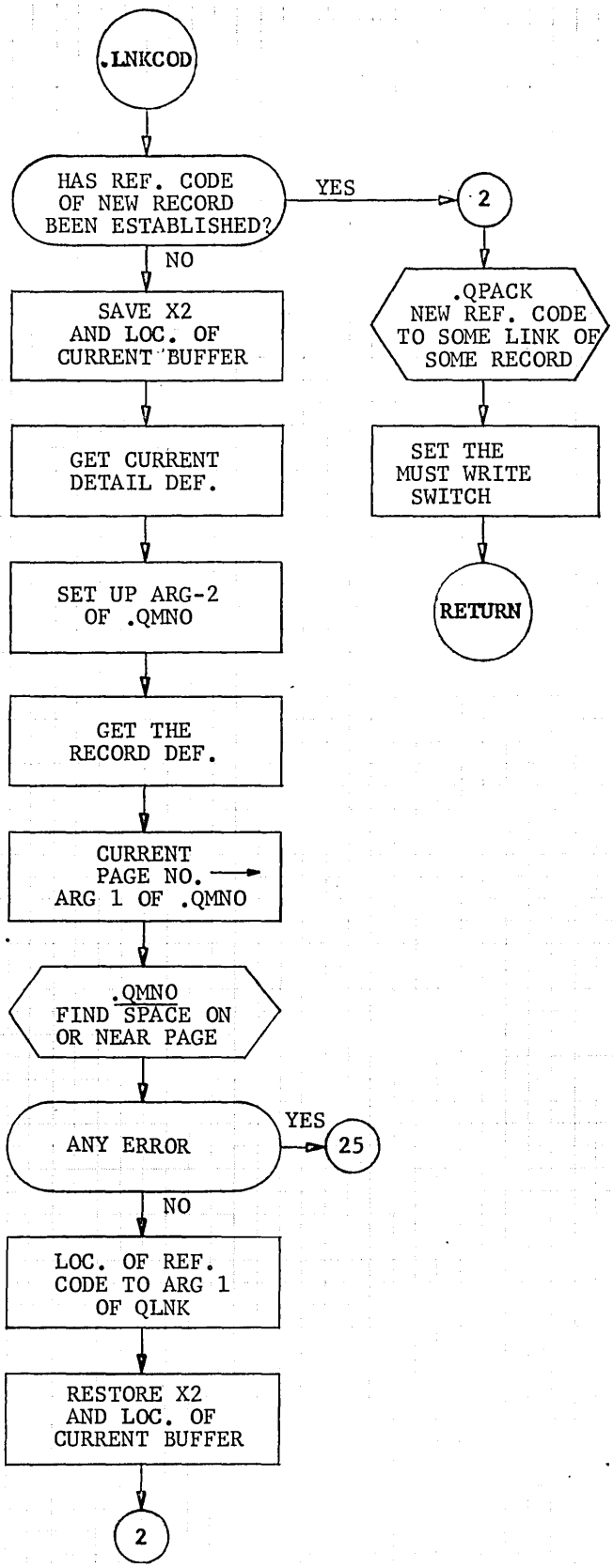
DETECTED ERRORS:

QUIT (58)	Attempt to link a record into a chain with the chain table NEXT equal to zero.
D01	Attempt to store an unallowed duplicate record.


.QLNK
SAVE X1, X3, X4
SAVE MASTER DEF.
IS CH. TBL. NEXT = 0?
NO
YES
ABORT
QUIT 58
CT NEXT = CURRENT i.e. NO DETAILS IN CHAIN?
NO
4
YES
QTYDEX
RETRIEVE CHAIN MASTER
ANY ERROR?
YES
25
NO
IS MASTER REC. PRIOR PROCESSABLE?
NO
14
YES
SET UP TO CHANGE PRIOR LINK OF MASTER
LNKCOD
REF. CODE TO PRIOR LINE
14
4
IS RELINK SWITCH SET?
YES
30
NO
DOES DETAIL LINK TO CURRENT RECORD?
YES
NO
BUILD ARG FOR QTYPX
31
.QFWD (4)
RETRIEVE MASTER
ANY ERROR?
YES
25
NO
SORTED CHAIN?
NO
32
5
YES
.QGDET
SEARCH FOR INSERT POINT
NO
ANY ERROR?
YES
25
NO
M.D., MASTER RETRIEVED?
YES
8
NO
D.D. DUPLICATE RETRIEVED?
YES
6
NO
GET D.D. BY HEADING CONTROL CHAIN
8
6
YES
ARE DUPLICATES ALLOWED?
NO
DO1 ERROR CODE TO C.C.B.
25
DUPLICATES ALLOWED FIRST?
NO
5
8
YES
IS RECORD PRIOR PROCESSABLE?
NO
12
YES
SET UP TO CHANGE PRIOR LINK OF RECORD
LNKCOD
REF. CODE TO PRIOR LINK
12
GET THE MASTER DEF.
QTYPEX, RETRIEVE C.T. PRIOR
ANY ERROR?
YES
50
NO
14


14
SET UP TO CHANGE NEXT LINK OF RECORD
LINKCOD REF. CODE TO NEXT LINK
GET THE MASTER DEFINITION
IS ORDER LINK BEFORE KEY?
NO
YES
CT CURRENT → CT NEXT
20
CT CURRENT REPLACES → CT PRIOR
CT CURRENT EQ. CT NEXT?
YES
20
NO
IS RECORD PRIOR PROCESSABLE?
NO
20
YES
.QTYPEX, RETRIEVE C.T. NEXT
ANY ERROR?
YES
50
NO
SET UP TO CHANGE PRIOR LINK OF RECORD
LNKCOD REF. CODE TO PRIOR LINK
GET THE MASTER DEF.
20
REF. CODE TO CH. TBL. CURRENT
.QTYPEX RETRIEVE NEW REC.
ANY ERRORS?
YES
50
NO
CH. TBL. NEXT → NEXT LINK OF NEW RECORD
IS RECORD LINKED TO MASTER?
NO
23
YES
CH. TBL. HEAD → HEAD LINK OF NEW RECORD
23
23
IS RECORD PRIOR PROCESSABLE?
NO
24
YES
CH. TBL. PRIOR → PRIOR LINK OF NEW RECORD
24
SET MUST WRITE SWITCH
26
RESTORE X1, X3, X4
RETURN


30
BUILD ARG FOR .QTYPX
31
.QTYPX RETRIEVE C.T. KEY
ANY ERRORS?
YES
25
NO
.QUDCH
32
DOES NEW REC. LINK BEFORE KEY?
YES
8
NO
CALC CHAIN DETAIL?
NO
14
YES
.QGDET LOOK FOR DUPLICATES
ANY ERROR?
YES
25
NO
MASTER DEF. RETRIEVED
NO
25
YES
33
33
.QFWD RETRIEVE CT NEXT
ANY ERROR?
YES
25
NO
GET THE MASTER DEF.
GET CHAIN TABLE NEXT
IS CT NEXT THE PAGE HEADER?
YES
14
NO
YES
IS CT NEXT SAME PAGE AS CT CURRENT?
NO
NO
IS SPACE AVAIL. ON PAGE OF CURRENT REC. SIZE?
YES
NO
IS SOME LINE NO. AVAIL. ON THIS PAGE?
YES
14
25
STORE ERROR CODE IN C.C.B.
SET CURRENT DET. DEF.
26
50
.QABORT FATAL ERROR


.LNKCOD
HAS REF. CODE OF NEW RECORD BEEN ESTABLISHED?
YES
2
NO
SAVE X2 AND LOC. OF CURRENT BUFFER
GET CURRENT DETAIL DEF.
SET UP ARG-2 OF .QMNO
GET THE RECORD DEF.
CURRENT PAGE NO. → ARG 1 OF .QMNO
.QMNO FIND SPACE ON OR NEAR PAGE
ANY ERROR
YES
25
NO
LOC. OF REF. CODE TO ARG 1 OF QLNK
RESTORE X2 AND LOC. OF CURRENT BUFFER
2
.QPACK NEW REF. CODE TO SOME LINK OF SOME RECORD
SET THE MUST WRITE SWITCH
RETURN

.QTYPX

FUNCTION:

Subroutine .QTYPX retrieves the specified record and compares its type with the types defined in the definitions of the chain chain, leaving in index register 2 the location of the definition corresponding to the record type retrieved. If the record retrieved is a CALC record undefined in the current environment, the first defined CALC record in sequence will be retrieved.

CALLING SEQUENCE:

CALL .QTYPX(ARG-1)

where ARG-1 is the location of the reference code of the record to be retrieved.

CALLED BY:

.QDLNK, .QHEAD, .QRUND, .QFWD, .QTLNK

CALLS FOR:

.QMRAC, .QPACK

LIMITATIONS AND RESTRICTIONS:

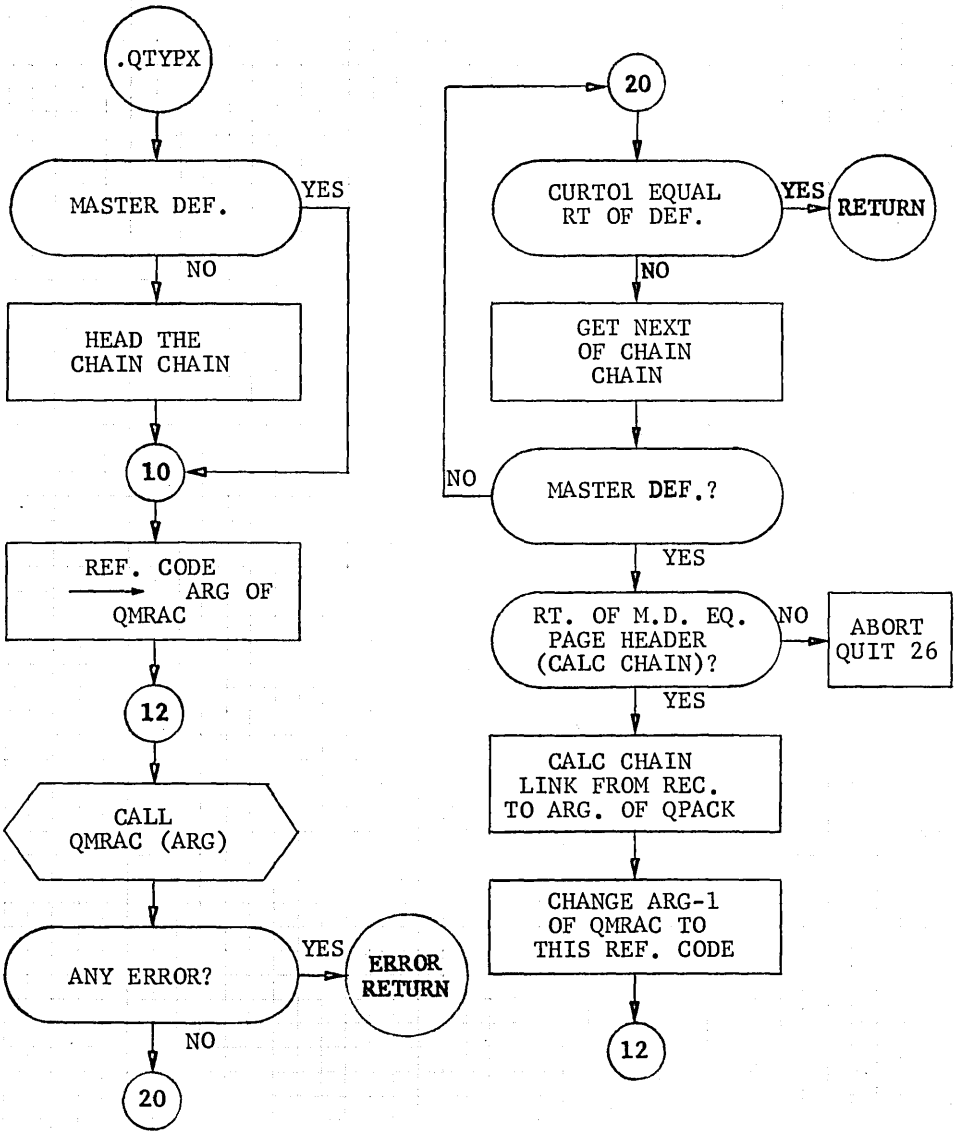
Index register 2 contains, at entry, the location of either a master definition or a detail definition designating the chain to which the required record belongs. At the normal exit, index register 2 contains the location of the master definition if the master record of the chain was retrieved; it contains the location of the detail definition if a detail record of the chain was retrieved.

NUMBER OF WORDS:

66 octal

DETECTED ERRORS:

QUIT (26) Record type of record is not defined for the chain specified.


.QTYPX
MASTER DEF.
YES
NO
HEAD THE CHAIN CHAIN
10
REF. CODE → ARG OF QMRAC
12
CALL QMRAC (ARG)
ANY ERROR?
YES
ERROR RETURN
NO
20
20
CURTO1 EQUAL RT OF DEF.
YES
RETURN
NO
GET NEXT OF CHAIN CHAIN
NO
MASTER DEF.?
YES
RT. OF M.D. EQ. PAGE HEADER (CALC CHAIN)?
NO
ABORT QUIT 26
YES
CALC CHAIN LINK FROM REC. TO ARG. OF QPACK
CHANGE ARG-1 OF QMRAC TO THIS REF. CODE
12

.QUCCB

FUNCTION:

Subroutine .QUCCB updates the communication control block with the current reference code and the current reference type. It also stores the location of the current record definition for use by other routines.

CALLING SEQUENCE:

CALL .QUCCB(ARG-1)

where ARG-1 is a word containing the reference code of the current record in bits 12-35.

CALLED BY:

.QCHN, .QGET, .QGETC, .QGETD, QGETE

CALLS FOR:

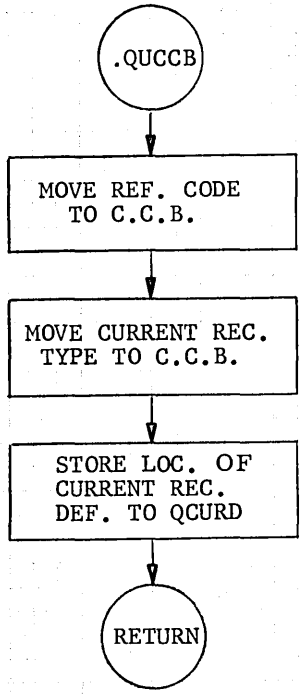
Not applicable

LIMITATIONS AND RESTRICTIONS:

Index register 2 must contain the location of the current record definition.

NUMBER OF WORDS:

15 octal


.QUCCB
MOVE REF. CODE TO C.C.B.
MOVE CURRENT REC. TYPE TO C.C.B.
STORE LOC. OF CURRENT REC. DEF. TO QCURD
RETURN

.QUDCH

FUNCTION:

Subroutine .QUDCH updates the chain table contained in the master definition for the chain specified. The chain table is updated with the chain pointers contained in the current record.

CALLING SEQUENCE:

CALL .QUDCH(ARG-1, ARG-2)

where ARG-1 is a word containing a zero or nonzero value. The update process varies depending on this value and on the types of pointers contained in the record. Normally, ARG-1 contains zero; however, when the chain table is being updated as a result of a "get next" process, ARG-1 contains a nonzero value.

ARG-2 is the location of a word containing the reference code of the current record in bits 12-35.

CALLED BY:

.QFWD, .QHEAD, .QTLNK, .QUPDC, .QDLTE

CALLS FOR:

.QPACK

LIMITATIONS AND RESTRICTIONS:

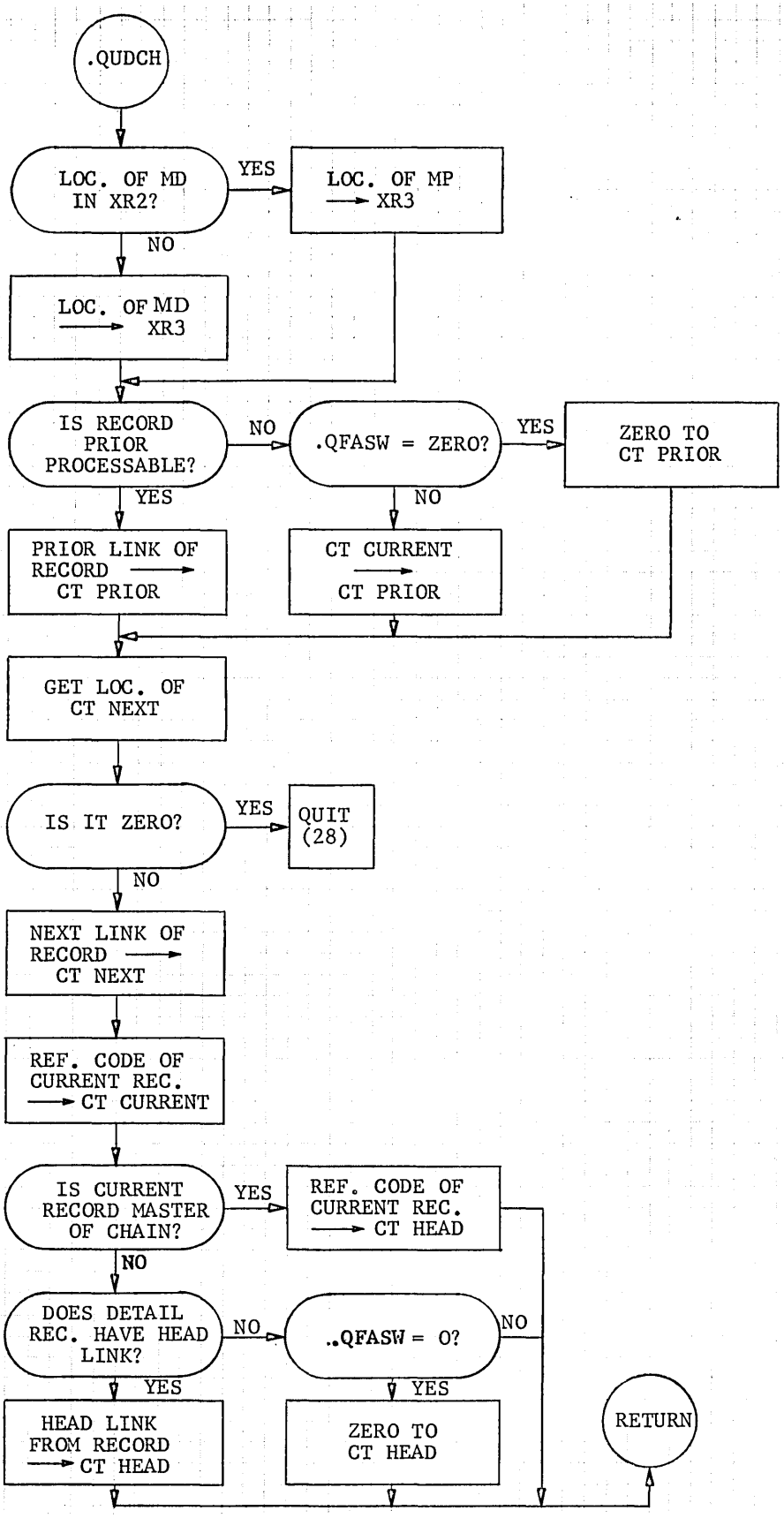
Index register 2 must contain the location of either a master or a detail definition designating the chain table to be updated.

NUMBER OF WORDS:

142 octal

DETECTED ERRORS:

QUIT (29) Position NEXT not defined for a given chain.


.QUDCH
LOC. OF MD IN XR2?
YES
LOC. OF MP → XR3
NO
LOC. OF MD → XR3
IS RECORD PRIOR PROCESSABLE?
NO
.QFASW = ZERO?
YES
ZERO TO CT PRIOR
YES
NO
PRIOR LINK OF RECORD → CT PRIOR
CT CURRENT → CT PRIOR
GET LOC. OF CT NEXT
IS IT ZERO?
YES
QUIT (28)
NO
NEXT LINK OF RECORD → CT NEXT
REF. CODE OF CURRENT REC. → CT CURRENT
IS CURRENT RECORD MASTER OF CHAIN?
YES
REF. CODE OF CURRENT REC. → CT HEAD
NO
DOES DETAIL REC. HAVE HEAD LINK?
NO
..QFASW = 0?
NO
YES
YES
HEAD LINK FROM RECORD → CT HEAD
ZERO TO CT HEAD
RETURN

.QUIT

FUNCTION:

Subroutine .QUIT is called in all cases in which abortion of the user program is appropriate. An abort code of D2 indicates an invalid use of the IDS functions or an invalid IDS definition structure. Abort code D1 indicates a hardware malfunction during an attempt to access the storage device. On any GECOS-or operator-initiated abort, the IDS file is closed and the program terminated. For IDS aborts D1 or D2, a message is printed on the SYSOUT execution report indicating the reason for the abort.

CALLING SEQUENCE:

CALL .QUIT(ARG-1)

where ARG-1 contains a reason code in binary.

CALLED BY:

.QASC, .QAUTH, .QBIC, .QCHN, .QCLOS, .QDLNK, .QDLTE, .QFWD, .QGDET, .QGET, .QGETC,.QGETD, .QGETE, .QHEAD, .QMDFY, .QMOVE, .QOPEN, .QSTOR, .QTLNK

CALLS FOR:

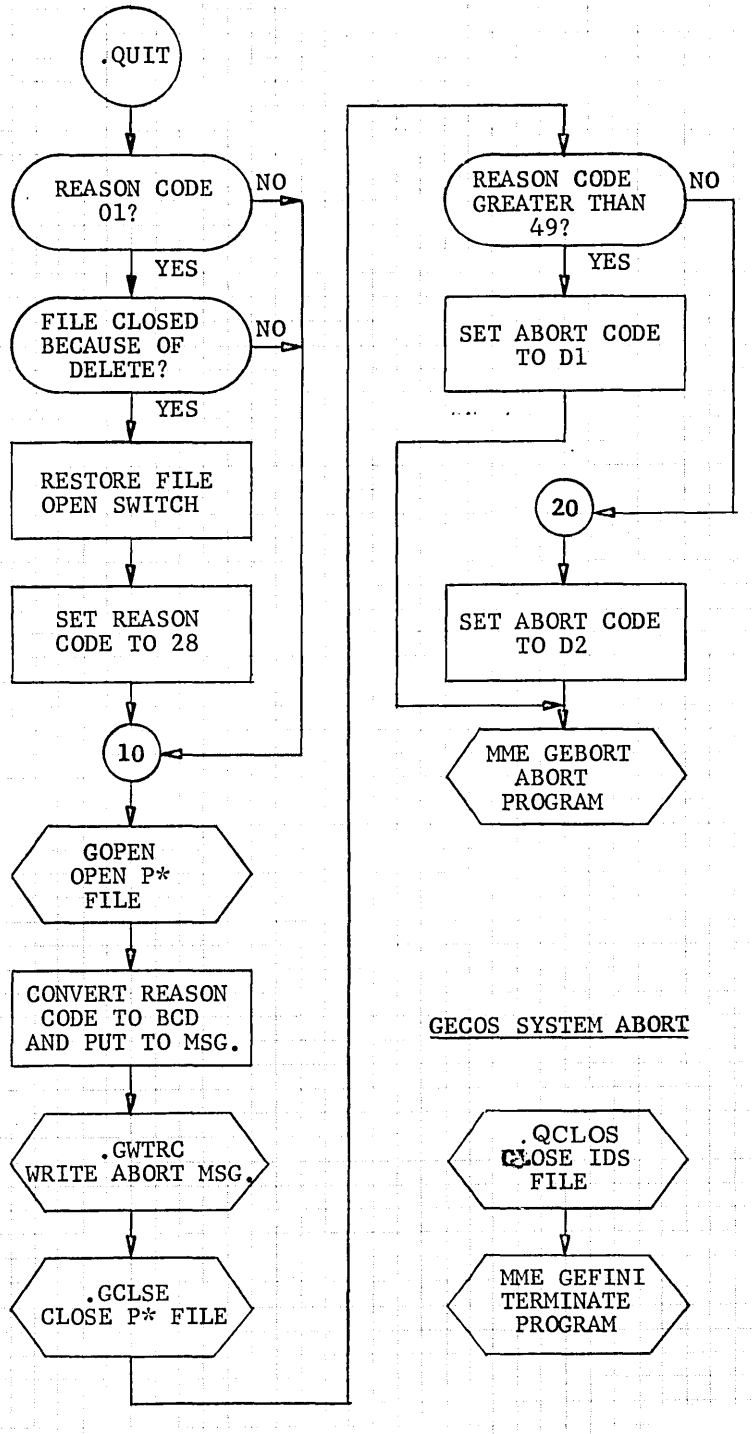
.GOPEN, .GWTRC, .GCLSE

NUMBER OF WORDS:

131 octal

DETECTED ERRORS:

QUIT (28)	Invalid use of an IDS statement within a conditional delete.


.QUIT
REASON CODE 01?
NO
YES
FILE CLOSED BECAUSE OF DELETE?
NO
YES
RESTORE FILE OPEN SWITCH
SET REASON CODE TO 28
10
GOPEN OPEN P* FILE
CONVERT REASON CODE TO BCD AND PUT TO MSG.
.GWTRC WRITE ABORT MSG.
.GCLSE CLOSE P* FILE
REASON CODE GREATER THAN 49?
NO
YES
SET ABORT CODE TO D1
20
SET ABORT CODE TO D2
MME GEBORT ABORT PROGRAM
GECOS SYSTEM ABORT
.QCLOS CLOSE IDS FILE
MME GEFINI TERMINATE PROGRAM

.QUPDC

FUNCTION:

Subroutine .QUPDC updates the chain tables affected by the processing of a given record. The chain table for a given chain is located in the master definition. If index register 2 contains at entry the location of a record definition, all chains in which the record is a master or a detail will be updated. If index register 2 contains the location of a master definition or a detail definition, all chains except the chain designated by the given definition will be updated. In either case, the record definition will also be updated with the reference code of the current record.

CALLING SEQUENCE:

CALL .QUPDC(ARG-1)

where ARG-1 is a word containing the reference code of the current record in bits 12-35.

CALLED BY:

.QASC, .QCHN, .QGETC, .QGETD, .QGETE, .QSTOR, .QDLTE, .QMDFY

CALLS FOR:

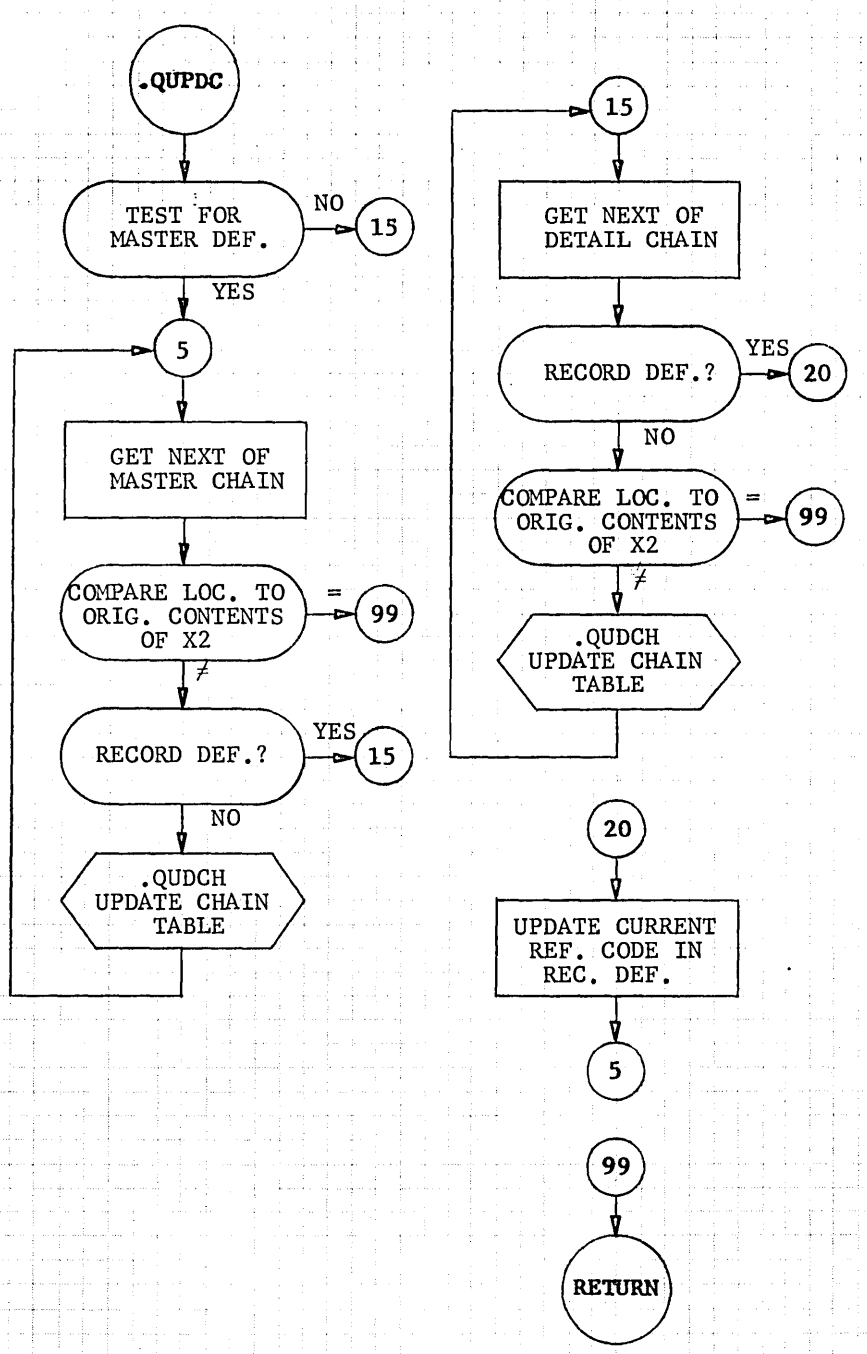
.QUDCH

LIMITATIONS AND RESTRICTIONS:

Index register 2 must contain the location of either a master, a detail, or a record definition.

NUMBER OF WORDS:

57 octal


.QUPDC
TEST FOR MASTER DEF.
NO
15
YES
5
GET NEXT OF MASTER CHAIN
COMPARE LOC. TO ORIG. CONTENTS OF X2
=
99
≠
RECORD DEF.?
YES
15
NO
.QUDCH UPDATE CHAIN TABLE
15
GET NEXT OF DETAIL CHAIN
RECORD DEF.?
YES
20
NO
COMPARE LOC. TO ORIG. CONTENTS OF X2
=
99
≠
.QUDCH UPDATE CHAIN TABLE
20
UPDATE CURRENT REF. CODE IN REC. DEF.
5
99
RETURN

.QUTLA

FUNCTION:

The IDS Utility Routine (CD600H5.135) enables the user to initialize N allocated storage devices, where N is 1 to 4 (maximum number of devices), with the required page header format used by IDS. In addition, the required space inventory page(s) will be appended to the pages used for data storage. An input data card is required with the following format:

Col. 1-6 number of characters per page, left-justified.
Col. 7-12 total number of pages in IDS file, left-justified.
Col. 13-24 12-character alphanumeric file name.

The following is a sample deck setup, with the appropriate control cards, to execute the initialization routine from the subroutine library (L*):

```
$   IDENT
$   USE         .QUTLA
$   ENTRY       .QUTLA
$   EXECUTE
$   LIMITS      4,6000,0,250
$   DISC or $   PERM if a permanent data file had been previously reserved by a Catalog
                Maintenance run.
$   DATA        I*
Input data card.
```

With the addition of the Space Inventory records (see Appendix A), the user must now be careful to supply the correct number of links (3840 words of storage on a DSU200 or drum) for his data file plus inventory pages.

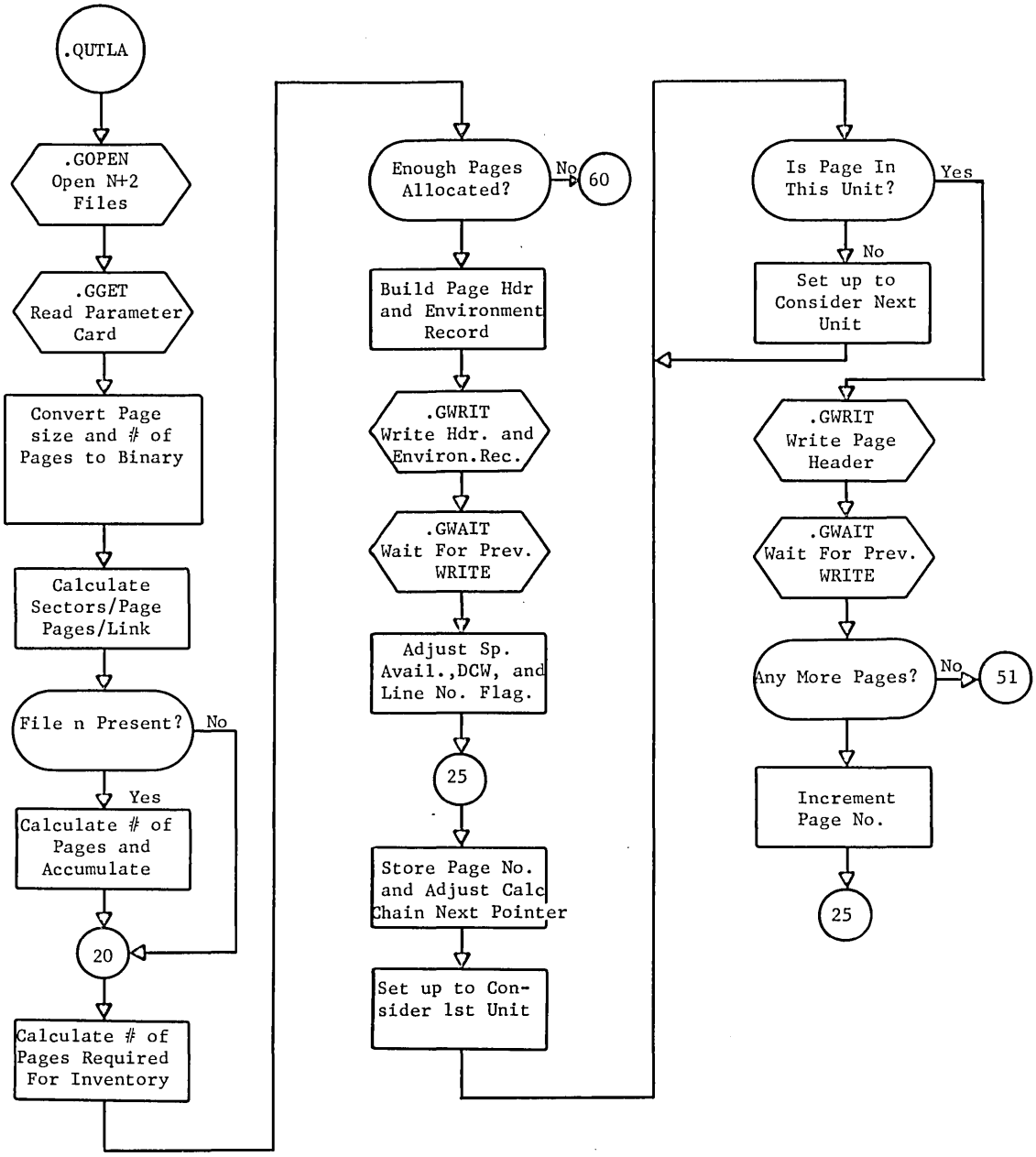
The following is a formula devised to aid the user in determining the total number of links required:

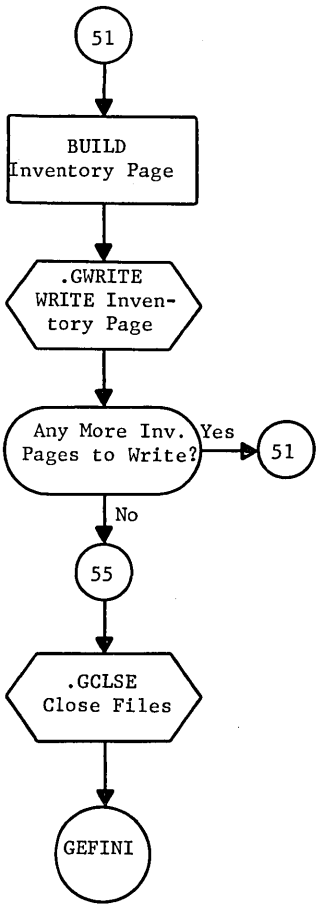
$$\text{Number of links} = \frac{N \times S + 1/3\ N}{S_I}$$

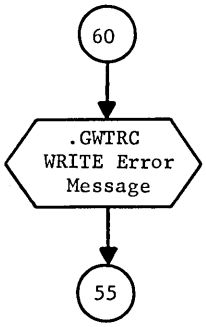
where:

N = Number of pages
S = Page size in words
S_I = Link size in words (3840)

All rounding is to the next highest integer.


.QUTLA
.GOPEN
Open N+2
Files
.GGET
Read Parameter
Card
Convert Page
size and # of
Pages to Binary
Calculate
Sectors/Page
Pages/Link
File n Present?
No
Yes
Calculate # of
Pages and
Accumulate
20
Calculate # of
Pages Required
For Inventory
Enough Pages
Allocated?
No
60
Build Page Hdr
and Environment
Record
.GWRIT
Write Hdr. and
Environ.Rec.
.GWAIT
Wait For Prev.
WRITE
Adjust Sp.
Avail.,DCW, and
Line No. Flag.
25
Store Page No.
and Adjust Calc
Chain Next Pointer
Set up to Con-
sider 1st Unit
Is Page In
This Unit?
Yes
No
Set up to
Consider Next
Unit
.GWRIT
Write Page
Header
.GWAIT
Wait For Prev.
WRITE
Any More Pages?
No
51
Increment
Page No.
25


51
BUILD
Inventory Page
.GWRITE
WRITE Inven-
tory Page
Any More Inv.
Pages to Write?
Yes
51
No
55
.GCLSE
Close Files
GEFINI


60
.GWTRC
WRITE Error
Message
55

.QUTLB

FUNCTION:

The Utility Routine--Storage Dump/Print (.QUTLB), CD600H5.136, dumps or prints the appropriate storage device(s) allocated to IDS. Both functions are optional and are controlled by the presence of an input data card.

The data card for the dump to tape has the following format:

Col. 1-5	TAPE

The data card for the dump to printer has the following format:

Col. 1-5	PRINT
Col. 13-17	Starting page number, left justified.
Col. 19-23	Ending page number, left justified. A 0 in Col. 13 will print the full file. Up to 10 print cards, in sequence by page number, may be entered at one time.

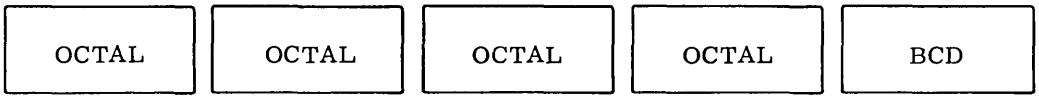
The following is a sample deck setup with the appropriate control cards needed to execute this Utility Routine from the Subroutine Library:

```
1               8                   16
$               IDENT
$               USE                 .QUTLB
$               ENTRY               .QUTLB
$               EXECUTE
$               LIMITS
$               DISC                                (or $ PERM)
$               TAPE                B1              (B1 is the required file code
                                                    for this tape)
```

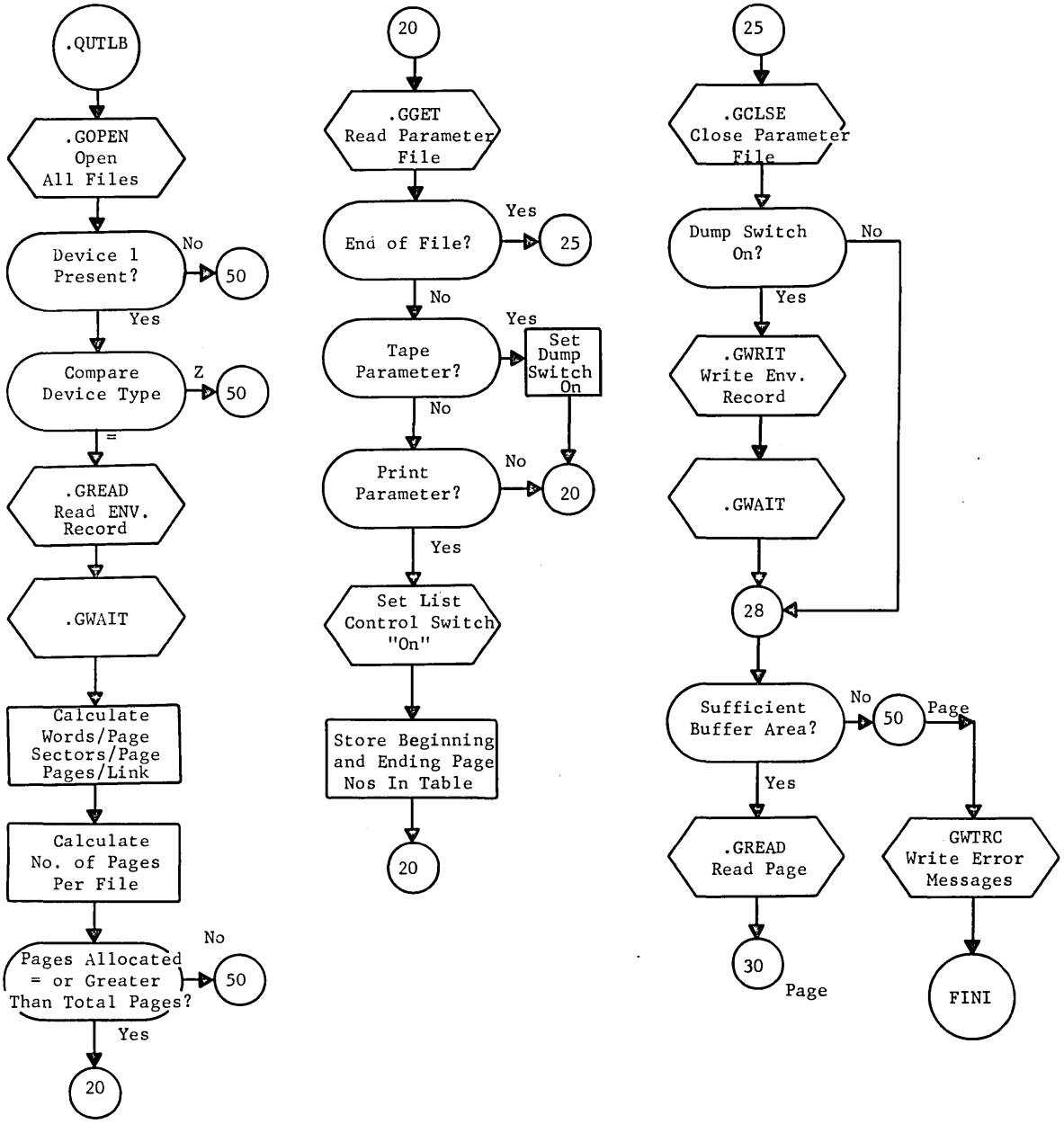
TAPE and or PRINT data cards.

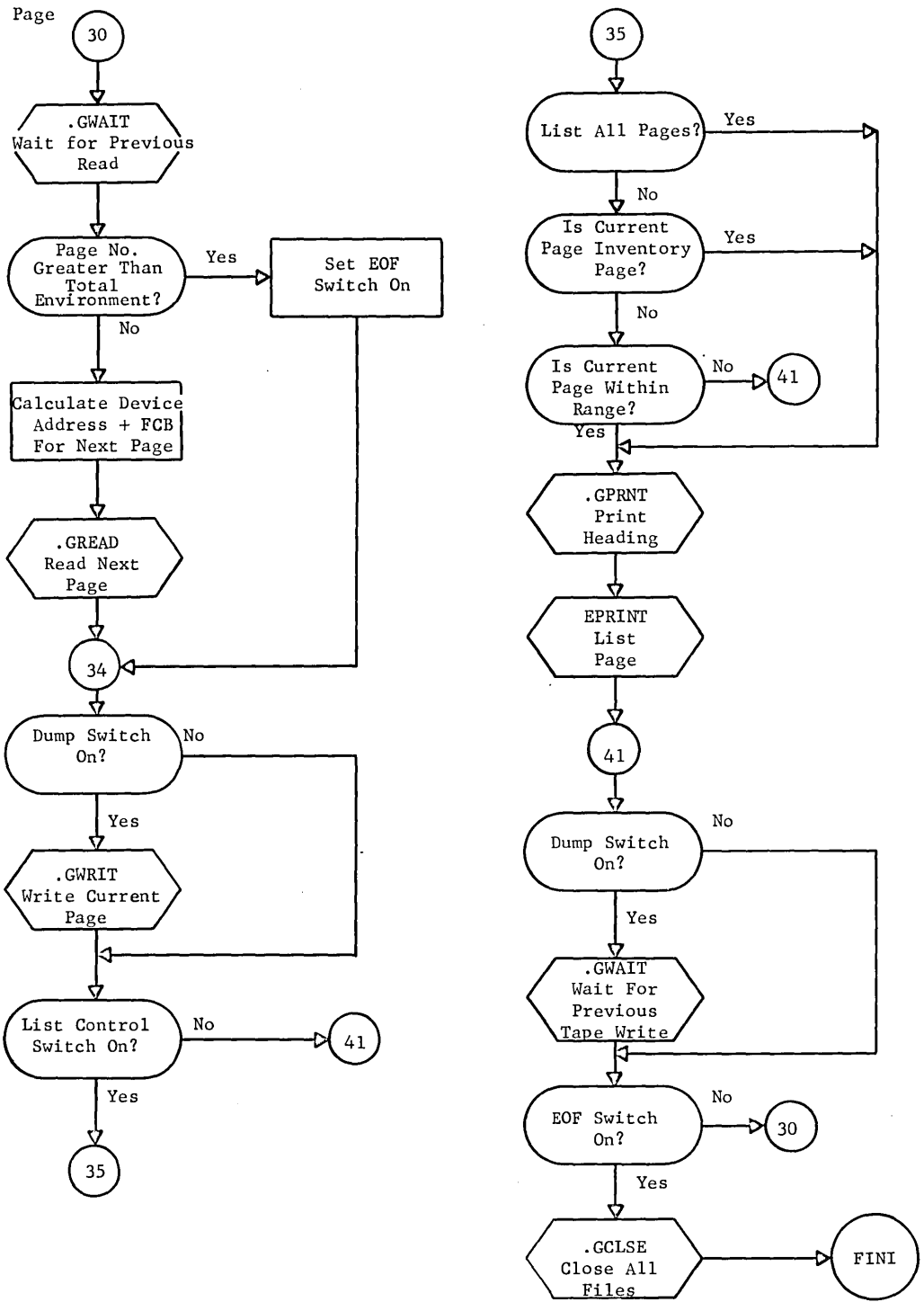
"PRINT" output from the dump routine is in the following format:

PAGE XXX ACTIVE PAGE SIZE XXXX



Side-by-side octal/BCD printout of the requested page.


.QUTLB
.GOPEN
Open
All Files
Device 1
Present?
No
50
Yes
Compare
Device Type
Z
50
=
.GREAD
Read ENV.
Record
.GWAIT
Calculate
Words/Page
Sectors/Page
Pages/Link
Calculate
No. of Pages
Per File
Pages Allocated
= or Greater
Than Total Pages?
No
50
Yes
20
20
.GGET
Read Parameter
File
End of File?
Yes
25
No
Tape
Parameter?
Yes
Set
Dump
Switch
On
No
Print
Parameter?
No
20
Yes
Set List
Control Switch
"On"
Store Beginning
and Ending Page
Nos In Table
20
25
.GCLSE
Close Parameter
File
Dump Switch
On?
No
Yes
.GWRIT
Write Env.
Record
.GWAIT
28
Sufficient
Buffer Area?
No
50
Page
Yes
.GREAD
Read Page
GWTRC
Write Error
Messages
30
Page
FINI


Page
30
.GWAIT
Wait for Previous
Read
Page No.
Greater Than
Total
Environment?
Yes
Set EOF
Switch On
No
Calculate Device
Address + FCB
For Next Page
.GREAD
Read Next
Page
34
Dump Switch
On?
No
Yes
.GWRIT
Write Current
Page
List Control
Switch On?
No
41
Yes
35
35
List All Pages?
Yes
No
Is Current
Page Inventory
Page?
Yes
No
Is Current
Page Within
Range?
No
41
Yes
.GPRNT
Print
Heading
EPRINT
List
Page
41
Dump Switch
On?
No
Yes
.GWAIT
Wait For
Previous
Tape Write
EOF Switch
On?
No
30
Yes
.GCLSE
Close All
Files
FINI

.QUTLC

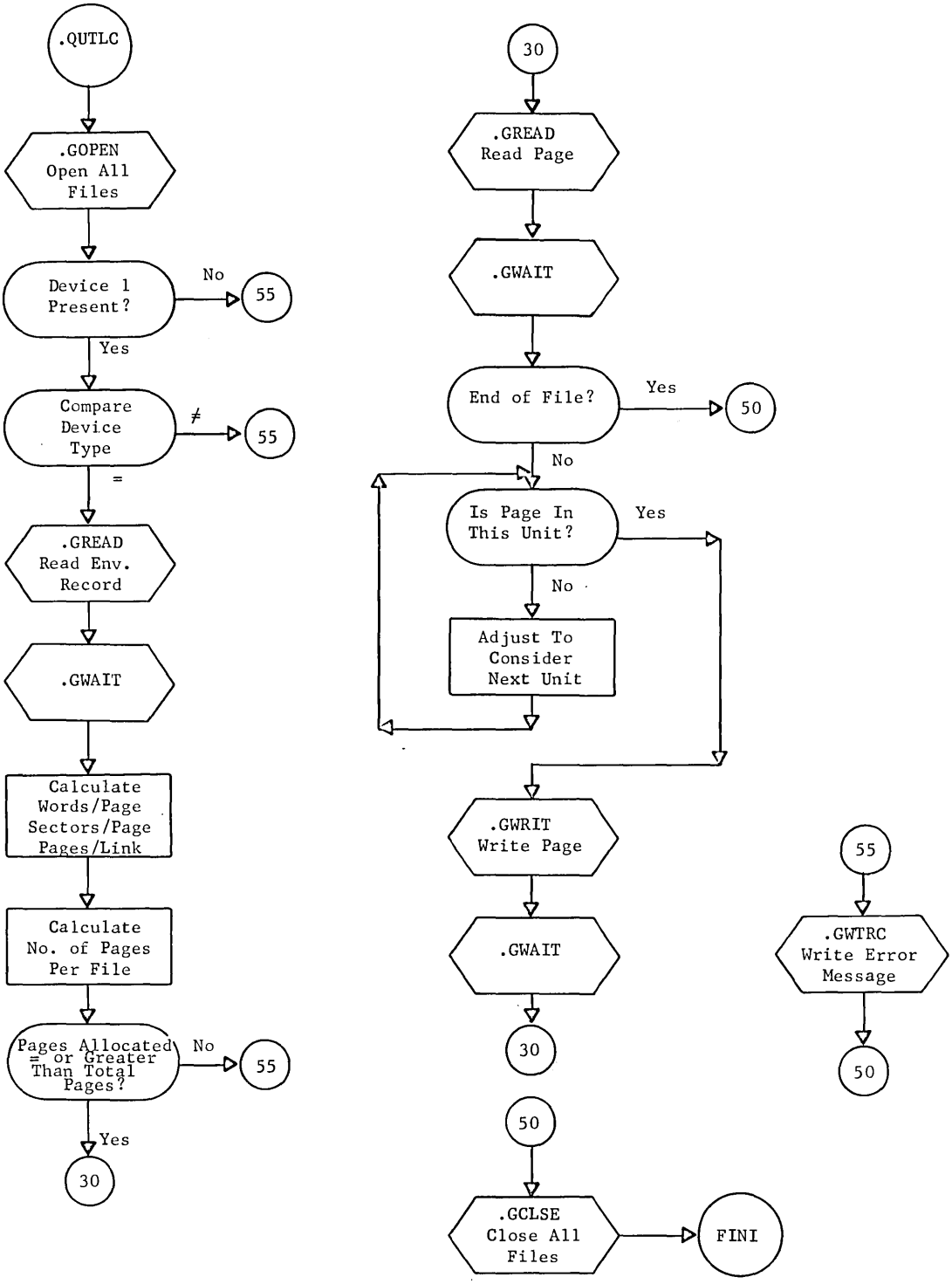
FUNCTION:

Update a mass storage device from a file created by the storage-to-tape routine or the Journal tape created by IDS.

The following is a sample deck setup with the appropriate control cards needed to execute this Utility Routine from the Sub-Routine library.

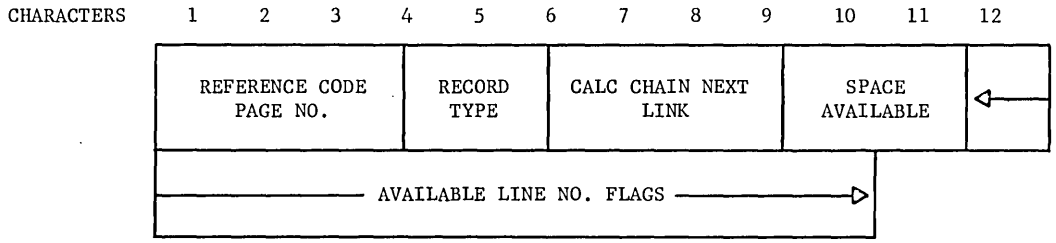
```
$       IDENT
$       USE         .QUTLC
$       ENTRY       .QUTLC
$       EXECUTE
$       LIMITS
$       DISC or $ PERM
$       TAPE        T1, Primary Logical Unit Designator,,,, File Name

                    where: T1 is the required
                    file code for this tape
```

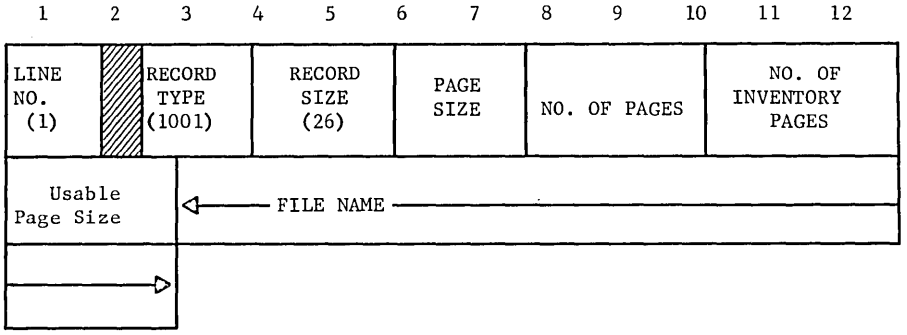

.QUTLC
.GOPEN
Open All
Files
Device 1
Present?
No
55
Yes
Compare
Device
Type
≠
55
=
.GREAD
Read Env.
Record
.GWAIT
Calculate
Words/Page
Sectors/Page
Pages/Link
Calculate
No. of Pages
Per File
Pages Allocated
= or Greater
Than Total
Pages?
No
55
Yes
30
30
.GREAD
Read Page
.GWAIT
End of File?
Yes
50
No
Is Page In
This Unit?
Yes
No
Adjust To
Consider
Next Unit
.GWRIT
Write Page
.GWAIT
30
50
.GCLSE
Close All
Files
FINI
55
.GWTRC
Write Error
Message
50

APPENDIX A
IDS RECORD FORMAT

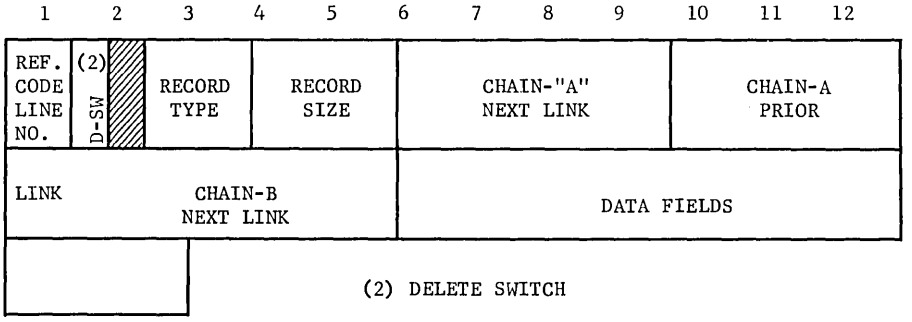
PAGE HEADER RECORD FORMAT



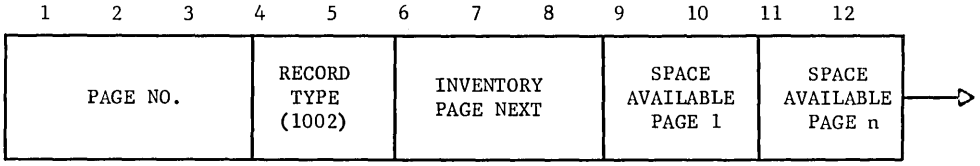
ENVIRONMENT RECORD



TYPICAL DATA RECORD FORMAT



INVENTORY PAGE



NOTE: Reference Code is a composite of the Page Numbers from the Page Header record and the unique Line Numbers from the data record.

APPENDIX B
IDS ERROR CONDITIONS

Two types of error conditions may occur during the execution of an IDS program. The first of these, involves those situations which are data dependent and must be anticipated by the procedural logic of the program. These errors are returned to the user program and may be tested by reference to a cell called ERROR-REFERENCE. The following codes are in this category.

R01	The retrieval of a record depends upon the selection of the current master of a given type. That record has not been retrieved or has been deleted.
R02	Record cannot be retrieved because the record or one of its masters has been deleted.
R03	The primary record retrieved is not the same record type as that specified in the Record Definition.
R04	Attempt to retrieve a record which does not exist within the data structure identified by supplied control fields.
R05	Attempt to retrieve the current record of a given type when that record type has not been previously retrieved or has been deleted.
R06	Attempt to retrieve DIRECT when DIRECT-REFERENCE is zero.
R07	Attempt to retrieve a record which has been deleted and its Delete switch is set.
R08	Reference code of the record to be retrieved is not found within the specified page.
R09	Reference code of the record to be retrieved is outside the total range of pages of the file.
D01	Attempt to store an unallowed duplicate record.
S01	Attempt to store a record when there is no space available within the range of pages specified for the record.

The second type of error involves those situations which are the result of improper use of the IDS functions, invalid definition of the data file, or hardware malfunctions which cannot be recovered by the software. These conditions, which are listed on the following page, result in an abort of the program with the following message printed on the execution report:

*** ABORTED BY IDS QUIT ROUTINE REASON CODE XX ***

In addition to the above message, a memory dump of the program is produced. Any time an abort occurs for any reason within an IDS program, the IDS file is first CLOSED with the appropriate IDS pages restored to the mass storage device.

Reason Code	Description
01	Attempt to store or retrieve a record with the IDS file not opened.
02	Capacity of the disc file allocated via GECOS is not large enough for the number of pages specified by the environment record.
03	Insufficient memory allocated to provide at least 2 buffers.
04	Authority-key does not match authority lock.
05	Invalid device-type allocation.
10	Definition Error-No detail definition for secondary or calculated record.
11	IDS subroutine table overflow. QASC, QDLTE and QSTOR must be reassembled.
12	Definition Error-Unique field for primary record has not been defined.
13	Definition Error-No record definition has been established for this record type.
14	Usage Error-Attempt to head chain for which no current record exists.
15	Attempt to Modify, Delete or Store a record with processing mode not equal to "Update."
16	Field to be modified or moved not defined for the current record.
17	Attempt to delete a record not previously retrieved.
18	Attempt to retrieve next in a chain for which no current record exists.
19	A control definition has been encountered which has an invalid control code.
20	A sort control or a randomize control field is specified with a record increment of zero.
21	Record definition of a record to be stored indicates that the record is less than 6 characters long.
22	Record definition of a record to be stored indicates that the record size is greater than page size 22.
23	Attempt to store a page header record.
24	Attempt to store a primary record with no unique field defined.
25	Attempt to store a record which is a detail in one or more chains when no storage chain is specified.

26	Record type of record retrieved is not defined for the chain specified.
27	Invalid conditional action specified for a conditional delete statement.
52-55	Linking of a record cannot be completed. Either the record cannot be retrieved from the disc or, for some other reason, the chain-pointer has been lost or is invalid.
56	Page read from mass storage device is not the page requested.
57	Attempt to delink a record when the chain table "NEXT" is zero. (Subroutine error)
58	Attempt to link a record into a chain with the chain table "NEXT" equal to zero. (Subroutine error)

DOCUMENT REVIEW SHEET

TITLE: GE-625/635 Integrated Data Store

CPB #: 1133A

FROM:

Name: _____

Position: _____

Address: _____

_____

Comments concerning this publication are solicited for use in improving future editions. Please provide any recommended additions, deletions, corrections, or other information you deem necessary for improving this manual. The following space is provided for your comments.

COMMENTS: _____

Please cut along this line

NO POSTAGE NECESSARY IF MAILED IN U.S.A.
Fold on two lines shown on reverse
side, staple, and mail.

FOLD

FIRST CLASS
PERMIT, No. 4332
PHOENIX, ARIZONA

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

POSTAGE WILL BE PAID BY

GENERAL ELECTRIC COMPANY
COMPUTER EQUIPMENT DEPARTMENT
13430 NORTH BLACK CANYON HIGHWAY
PHOENIX, ARIZONA - 85029

ATTENTION: ENGINEERING PUBLICATIONS STANDARDS B-90

FOLD

Progress Is Our Most Important Product

GENERAL ELECTRIC

INFORMATION SYSTEMS DIVISION

LITHO U.S.A.