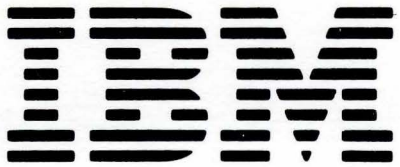


SC21-7806-4

File No. S38-36

IBM System/38

IBM System/38
Control Program Facility
Reference Manual–
Data Description Specifications

Program Number 5714-SS1

SC21-7806-4

File No. S38-36

IBM System/38

IBM System/38
Control Program Facility
Reference Manual–
Data Description Specifications

Program Number 5714-SS1

Fifth Edition (December 1981)

This is a major revision of, and makes obsolete, SC21-7806-3. See *About This Manual* for a summary of major changes to the previous edition. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change or addition.

This edition applies to release 3, modification 0 of the IBM System/38 Control Program Facility (Program 5714-SS1) and to all subsequent releases until otherwise indicated in new editions or technical newsletters. Changes are periodically made to the information herein; these changes will be reported in technical newsletters or in new editions of this publication.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Use this publication only for the purposes stated in *About This Manual*.

Publications are not stocked at the address below. Requests for copies of IBM publications and for technical information about the system should be made to your IBM representative or to the branch office serving your locality.

This publication could contain technical inaccuracies or typographical errors. Use the Reader's Comment Form at the back of this publication to make comments about this publication. If the form has been removed, address your comments to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901. IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Contents

PURPOSE OF THIS MANUAL

This manual is written for programmers who use IBM System/38.

This manual contains detailed instructions for coding the data description specifications (or DDS) for files that can be described externally. These files are the physical, logical, display, printer, communications files, and BSC files. Although Chapter 1 does review the process of creating such files, and Appendix F does provide examples of each type of file discussed in this manual, you should not read or attempt to use this manual without knowing how you intend to use externally described files in your application.

For an understanding of the context in which DDS is used, see the *CPF Concepts Manual* and the *CPF Programmer's Guide*. (The full titles and order numbers of these manuals appear at the end of this section.)

For detailed descriptions of the specific commands used to create and delete files or to change and override their attributes, seethe *CPF Reference Manual—CL*.

If your application involves data communications with another system, see the *Data Communications Programmer's Guide* before using Chapter 6, *Communications Files*.

If you are using program-described files, see the *CPF Programmer's Guide* and the appropriate high-level language manual.

ORGANIZATION OF THIS MANUAL

See the *Summary of Changes* later in this section for the most recent changes to this manual.

Chapter 1 contains an overview of DDS, describing:

- The function of DDS
- How you use DDS
- Types of files (that is, data base and device files)
- Three steps to creating an externally described file
  - Filling in the Data Description Specifications form
  - Entering DDS source statements
  - Creating files
- Describing files level by level
- Syntax coding examples
- DDS syntax rules

Chapters 2 through 6 discuss both positional entries and keyword entries for each type of file:

- Data base files
  - Physical files (Chapter 2)
  - Logical files (Chapter 3)
- Device files
  - Display files (Chapter 4)
  - Printer files (Chapter 5)
  - Communications files and BSC files (Chapter 6)

In the *Keyword Entries* sections of Chapter 4, *Display Files* and Chapter 6, *Communications Files*, sections called *Use of Routing Keywords* precede the first routing keyword (RTGAID). These sections explain some of the interrelationships among routing keywords.

The appendixes in this manual contain information on how to specify the REF and REFFLD keywords, keyword summaries for the various types of files, sample files, and a list of abbreviations used in DDS.

The *Glossary* follows the appendixes.

This manual uses the following conventions:

- *He* means *he* or *she*.
- In the keyword descriptions, *this field* or *this record format* means the field you are now defining or the record format you are now defining.
- The expression *use this file or record level keyword* means the keyword is valid only at the file or record level.
- *To specify a keyword* means to code the keyword in the DDS for a file. This contrasts with *to select a keyword* or *when a keyword is in effect*, which both mean that any conditioning (such as one or more option indicators) is satisfied when an application program issues an output or input operation (write or read operations).
- The mention of one option indicator in the keyword descriptions indicates that any valid number or combination of indicators can be specified.
- *Current source* or *source you are defining* means the DDS that together make up the description of one file.
- In sample displays, character fields are shown as all Xs and numeric fields are shown as all Ns.

SUMMARY OF CHANGES

The major changes to this publication include:

- New display file keywords as follows:
  - CLRL (clear line)
  - CSRLOC (cursor location)
  - OVRATR (override attribute)
  - OVRDTA (override data)
  - PUTOVR (put override)
  - RTNDTA (return data)
  - SLNO (starting line number)
- New optional parameter values for the following keywords:
  - ERASEINP
  - MDTOFF
  - UNLOCK
- New keywords and new keyword spellings for compatibility with other systems as follows:
  - New CHECK codes providing the same function as AUTO and LOWER
  - CMP also spelled as COMP
  - SETOF also spelled as SETOFF
  - Alternative parameter values can now be specified for DSPSIZ
- Extensive rewriting of the manual, with many new explanations and examples, including the following:
  - Complete rewriting of the field reference function
  - Complete rewriting of FORMAT and UNIQUE in data base files
  - Complete rewriting of *Key Field Names* in Chapter 3
  - Thorough revision of *Data Type/Keyboard Shift* in Chapter 4
  - Elimination of *Use of Subfile Keywords*; information now in the CPF Programmer's Guide and in the SFL and SFLCTL keyword descriptions
  - Complete rewriting of many keywords in Chapter 4, including all the subfile keywords
  - Thorough revision of communications file keywords and addition of BSC information
  - New Combined Chart for handy reference to display file keywords in Appendix C
  - New subfile examples in Appendix F
- Miscellaneous technical changes indicated by a vertical line to the left of the change or addition

WHAT YOU SHOULD KNOW

To use this manual effectively, you should know how to:

- Use SEU or data-entry system to enter DDS source
- Enter the create file commands on System/38 (such as Create Physical File [CRTPF] or Create Display File [CRTDSPF])
- Use a high-level language (RPG III, COBOL, or CL) to develop an application program

IF YOU NEED MORE INFORMATION

To correct a problem, you may need to refer to another IBM publication for a specific type of information:

CPF (Control Program Facility) Commands and Functions

- *IBM System/38 Control Program Facility Concepts Manual*, GC21-7729
  - Describes the concepts involved in data base design and application development

- *IBM System/38 Control Program Facility Programmer's Guide*, SC21-7730
  - Describes, and contains examples of, using DDS to describe data base and device files
  - Describes the processing of data base files
  - Describes physical and logical file members
  - Describes using display files in a program
  - Describes the I/O requests that can be made to a display file
  - Describes the device-dependent attributes of card, diskette, tape, and printer files, and describes how to use the files
  - Contains a description of the data management feedback area
  - Describes creating and changing source files
  - Describes copying to and from source files
  - Describes using an inline data file
  - Describes processing considerations for inline data files and spooled output files
  - Describes how to create a message file and user-defined messages

- *IBM System/38 Control Language Reference Manual*, SC21-7731
  - Describes control language syntax
  - Describes control language commands and parameters
  - Lists command authorizations by user profile

- *IBM System/38 Programmer's/User's Work Station Guide*, SC21-7744
  - Describes how to enter commands, using the command entry display and the prompting facilities provided by CPF
  - Describes how to communicate with System/38 through an interactive work station, especially when using the functions available through the Control Program Facility (CPF)

- *IBM System/38 Application Example I*, SC21-7881
  - Describes how to use spooling for input of source statements
  - Describes how to create programs from diskette input (the same process is used to create files from diskette input)
  - Describes how to use the source entry utility (SEU) to enter source for files and programs
  - Describes how to use the programmer menu to create files and programs
  - Describes how to create and execute a data file utility (DFU) application
  - Describes how to create a menu from a separate display file and control language program
  - Describes how to create a menu using the screen design aid (SDA)
  - Describes how to use a logical file to create a different access path

Languages

This list includes the publications referred to as *appropriate high-level language manual* in this manual:

- *IBM System/38 Introduction to RPG III: External Data Description and Interactive Processing*, GC21-7723
  - Describes how to code an interactive RPG program that prints a report
  - Describes how to code an interactive RPG program that inquires into an externally described file
  - Describes how to enter, compile, and run the program
- *IBM System/38 RPG III Reference Manual and Programmer's Guide*, SC21-7725
  - Describes how to code RPG III programs
  - Describes how to compile and run RPG III programs
  - Describes how to test and debug RPG III programs
  - Describes how to interpret compiler listings and messages
  - Describes how to perform problem determination procedures
- *IBM System/38 Concepts for the COBOL User*, GC21-7855
  - Contains conceptual information on the System/38 data base and work station environment
  - Relates COBOL language functions to system functions
- *IBM System/38 COBOL Reference Manual and Programmer's Guide*, SC21-7718
  - Describes how to code COBOL programs
  - Describes how to interpret COBOL compiler listings and messages
  - Describes how to test and debug COBOL programs
  - Describes how to perform problem determination procedures
- *IBM System/38 Control Program Facility Programmer's Guide*, SC21-7730
  - Describes, and contains examples of, control language programs
  - Describes how to use variables, display files, and messages in control language programs
  - Describes how to monitor for messages in control language programs

Utilities

- *IBM System/38 Screen Design Aid Reference Manual and User's Guide*, SC21-7755
  - Describes how to design, create, maintain, and test display files interactively (using SDA)
- *IBM System/38 Source Entry Utility Reference Manual and User's Guide*, SC21-7722
  - Describes how to enter and maintain DDS source statements using SEU
  - Describes how to browse through a spooled file to see compiler output, scan for errors, and correct source statements that contain errors
- *IBM System/38 Data File Utility Reference Manual and User's Guide*, SC21-7714
  - Describes how to enter data records into a physical file using DFU
  - Describes how to change or delete data records in a physical file using DFU

Messages

- *IBM System/38 Messages Guide: CPF, RPG III, and IDU*, SC21-7736
  - Contains an overview of messages and the circumstances under which they may appear
  - Describes the elements of messages
  - Contains a problem determination section, which includes additional information about the problem for which the message was issued
  - Contains all messages other than COBOL messages
- *IBM System/38 Messages Guide: COBOL*, SC21-7823
  - Contains a detailed description of message structure and the meaning of each message element
  - Explains how to add user-defined messages to the system
  - Lists all COBOL messages with additional information that is not displayed or printed by the system

Coding and Debugging Material

- *IBM Data Description Specifications*, GX21-7754
  - To code (specify) DDS
- *IBM Data Description Specifications (DDS) Debugging Template*, GX21-7717
  - To interpret fields on the DDS source listings

Communications

- *IBM System/38 Data Communication Programmer's Guide*, SC21-7825
  - Describes the use of data description specifications (DDS) for communications; also, describes the Create Communications File (CRTCMNF) and Create BSC File (CRTBSCF) commands
  - Identifies communications programming considerations and contains examples of RPG III and COBOL programs for communications with BSC and SNA devices and systems
  - Describes error handling for communications
  - Identifies the device-dependent considerations for supported SNA hosts, and BSC devices and systems

Device Operation

- *IBM System/38 Programmer's/User's Work Station Guide*, SC21-7744
  - Describes the keys on the 5250 work station keyboards (typewriter-like and data-entry)
  - Describes the keys on the system console keyboard

Content and Use of System/38 Publications

- *IBM System/38 Guide to Publications*, GC21-7726
  - Describes contents of System/38 manuals
  - Describes reading sequences for System/38 manuals
- *IBM System/38 Glossary and Master Index*, GC21-7727
  - Defines terms used in System/38 manuals
  - Combines entries from indexes of System/38 manuals used frequently

Chapter 1. Introduction

THE FUNCTION OF DATA DESCRIPTION SPECIFICATIONS

A traditional means of describing data attributes (such as the names and lengths of records and fields) is to specify the data attributes in the application programs themselves. In RPG this is done in the input or output specifications, and in COBOL this is done in the Data Division. On System/38 this method of data description is called *program-described* data; however, a convenient and powerful alternative is available. Through the use of data description specifications (DDS), the programmer can describe data attributes *externally*; that is, not in the application programs themselves but in file descriptions independent of the programs. These file descriptions are associated with the files themselves. The files for which DDS can be used are:

- Physical files (DDS is optional)
- Logical files (DDS is required)
- Display files (DDS is recommended)
- Printer files (DDS is optional)
- Communications files and BSC files (DDS is required)

For card files, diskette files, and tape files, program-described data must be used.

The *CPF Reference Manual—DDS* is a reference manual for programmers using externally described files. For information about the context within which to use DDS, and for a description of program-described files, see the *CPF Programmer's Guide* and the appropriate high-level language manual.

CPF SUPPORT FOR DATA DESCRIPTION SPECIFICATIONS

The Control Program Facility (CPF), through data base data management and device data management, provides support for DDS at three points:

- When you create a file (with a create file command such as CRTLF and CRTDSPF). In this manual, this is called file creation time.
- When you compile an application program (with a create program command such as CRTRPGPGM and CRTCBLPGM). In this manual, this is called program compilation time.
- When your program issues a request to a file. In this manual, this is called execution time. Execution time includes the following:
  - When the file is opened or closed.
  - When your program issues an I/O operation to the file, specifically:
    a. Write or Put operations (in this manual, output operations)
    b. Read or Get operations (in this manual, input operations)
    c. Update operations

At file creation time for externally described files, CPF performs a function much like the compilation of application programs, invoking the data description processor, reading the DDS source, printing any diagnostic messages, and creating a CPF object (type *FILE). Further, CPF provides a reference function whereby record formats or field descriptions can be duplicated from an already existing file into the new file. CPF also establishes access paths as required.

At program compilation time, the language compiler brings data descriptions from externally described files into the executable program and lists the data descriptions on the compiler listing.

At execution time, CPF processes data passing between the application program and the data base or device.

See Figure 1-1 for an illustration of file access through DDS.

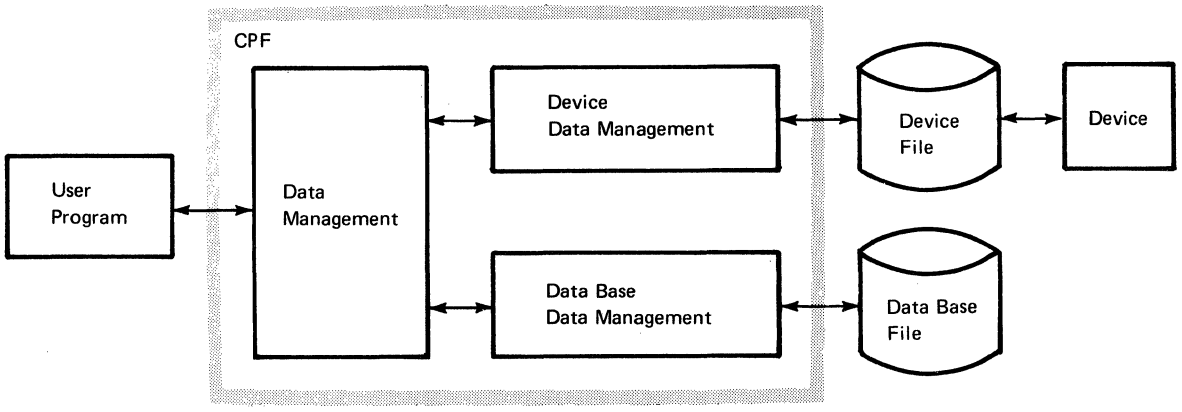


Figure 1-1. File Access at Execution Time

TYPES OF FILES

Two types of files are used to pass data between the System/38 and application programs:

- *Data base files* include physical files (which can actually contain data records) and logical files (which can contain only record formats, not data records, and which provide logical views of the data records stored in physical files).

- *Device files* include display files, printer files, communications files, and BSC (Binary Synchronous Communications) files. These types of file can be described using DDS. (Other types of device file, such as diskette, card, and tape files, cannot be described using DDS.) As illustrated in Figure 1-1, device files describe devices to the system so that device data management can pass data between your programs and the devices.

Data Base Files

Data base files describe how your data is stored and how it is presented to your programs. There are two types of data base files:

- *Physical files*: The DDS for a physical file describes how your data records are actually stored in the data base. Fields within the record format of a physical file determine the attributes of the fields in the data base. The presence of a key field indicates keyed sequence in the data base; when no key field is specified, arrival sequence is used in the data base.

- *Logical files*: The DDS for a logical file describes how data records appear to be stored in the data base; however, the logical file does not actually contain data records. A logical file is always based on one or more physical files. The logical file determines how data records are sequenced and transformed when read by an application program from the physical file(s) and how data records are transformed and sequenced when written from an application program to the physical file(s).

Figure 1-2 illustrates the relationship between two logical files (logical file A and logical file B) and a common physical file. Program A and program B see only the logical views of the data records as these views are presented by the logical files A and B. The DDS for the physical file describes how the data records are actually stored in the data base.

Figure 1-3 illustrates the relationship between one logical file and two physical files on which it is based. The program can access data in the two physical files through a record format in the logical file. Even if the physical files have different record formats or different sequences of records, to the program they are the same.

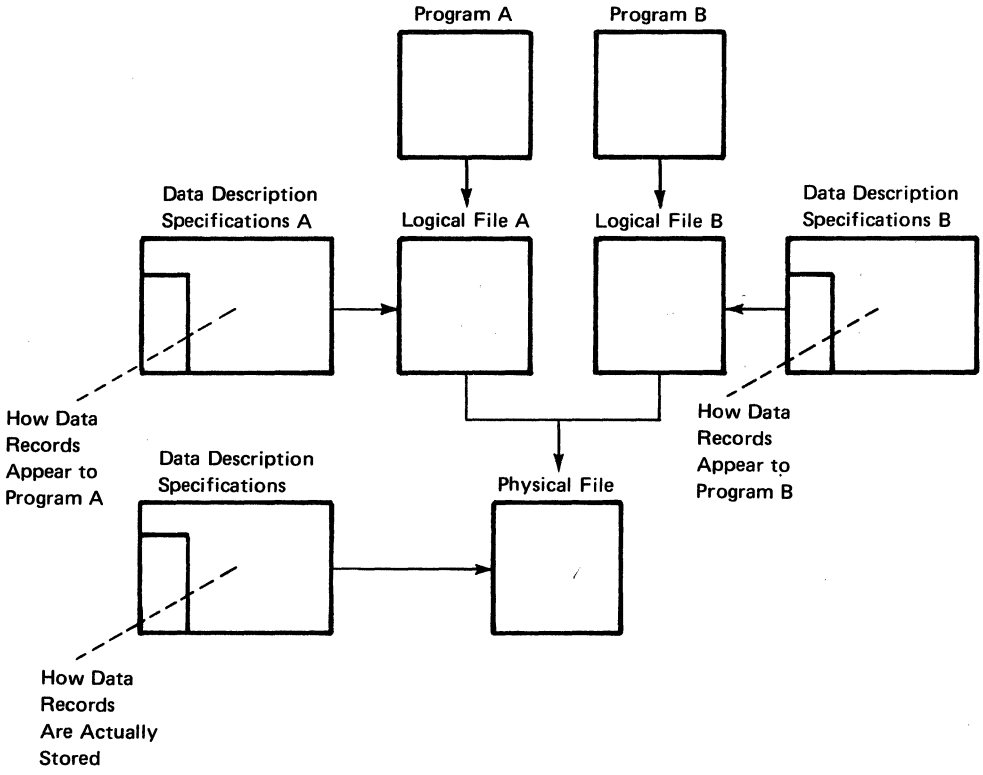


Figure 1-2. Two Logical Files Based on One Physical File

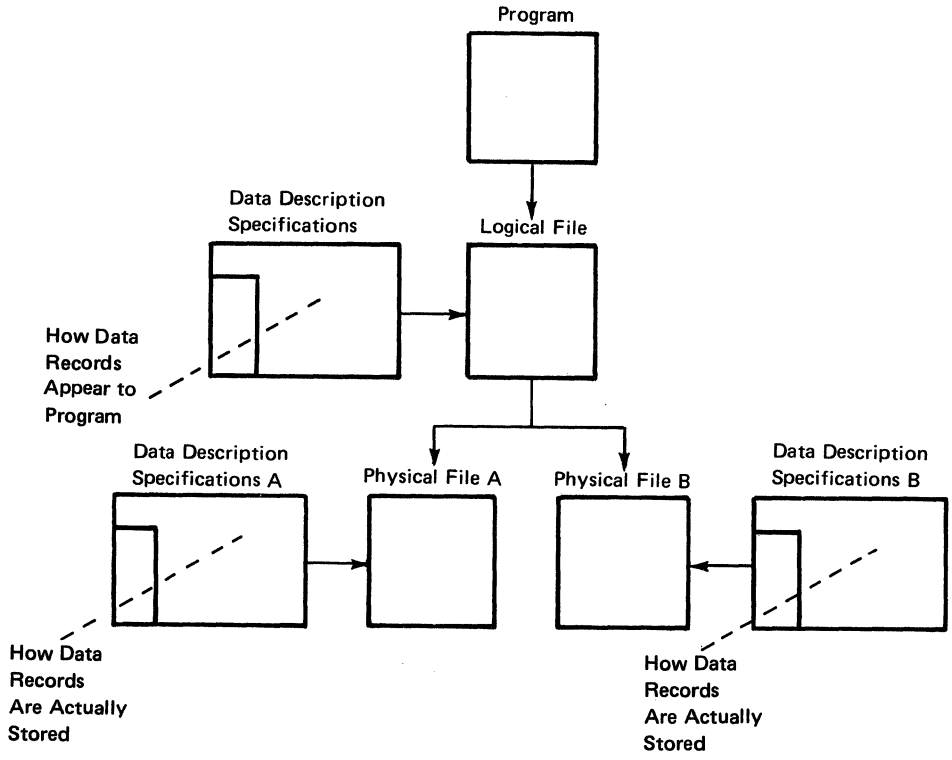


Figure 1-3. One Logical File Based on Two Physical Files

Coding Characteristics of Data Base Files

Physical files must have only one record format.

Logical files can have more than one record format, and each record format must have the PFILE keyword specified for it.

Field Reference Files

A field reference file is a physical file that defines all fields needed for an application or a group of related files. The field reference file contains no data and should be created without a member (MBR(*NONE) specified for the Create Physical File (CRTPF) command). Like all physical files, a field reference file has only one record format. As needed, you can define up to 8000 fields on the field reference file to form a standardized list of named files to be referenced by other files.

To reference fields in a field reference file, use the reference function provided by DDS. To use the reference function in a file you are defining, you specify R in position 29 for each field that references another field (see *Position 29 (Reference)* in the appropriate chapter for details) and the name field reference file as a parameter value on the REF or REFFLD keyword.

Certain field attributes, such as location, are required in device files but are invalid in data base files. When you reference a field reference file from a device file, you must specify the field attributes required by the device file. If you specify a field attribute that is already defined in the field reference file, you override the referenced attribute. Some such overrides affect other attributes; see *Position 29 (Reference)* for the effect overriding certain attributes has on others.

Access Paths

Each member of a data base file has an *access path*. The access path allows data management to sequence the records in the member. There are two types of access paths: arrival sequence and keyed sequence.

- *Arrival sequence access path*: Data management sequences the records of a member either sequentially or by relative record number. No key fields are specified in the DDS.

- *Keyed sequence access path*: Data management sequences the records of a member according to the contents of the key fields of the records. If you want keyed access to the records, you must specify key field information when you specify the DDS for the file. (See the sections on *Key Field Name* in Chapters 2 and 3 in this manual.)

Figure 1-4 shows how arrival sequence and keyed sequence access paths function. For a discussion of the considerations for access paths and data base file members, see the *CPF Programmer's Guide*.

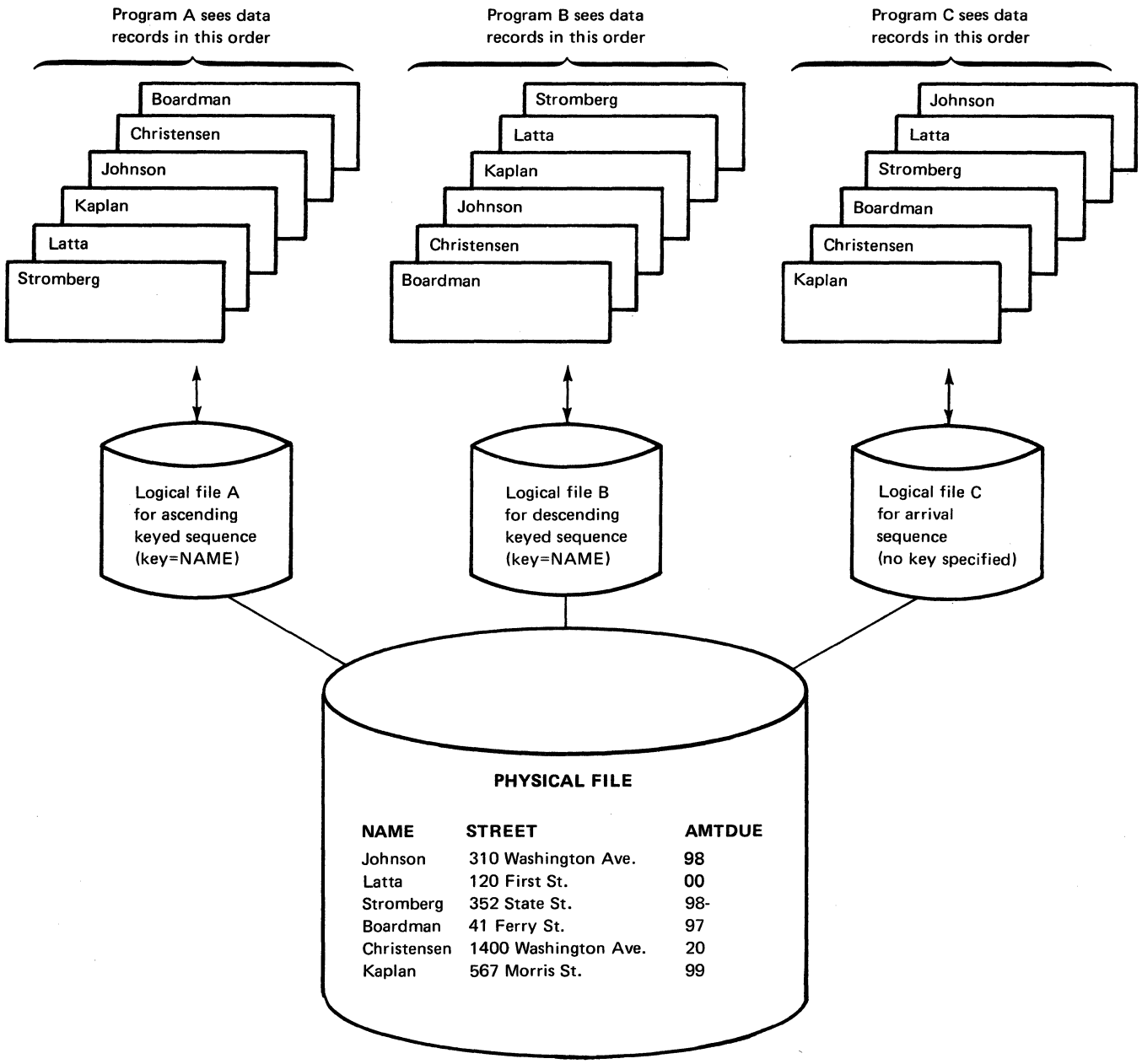
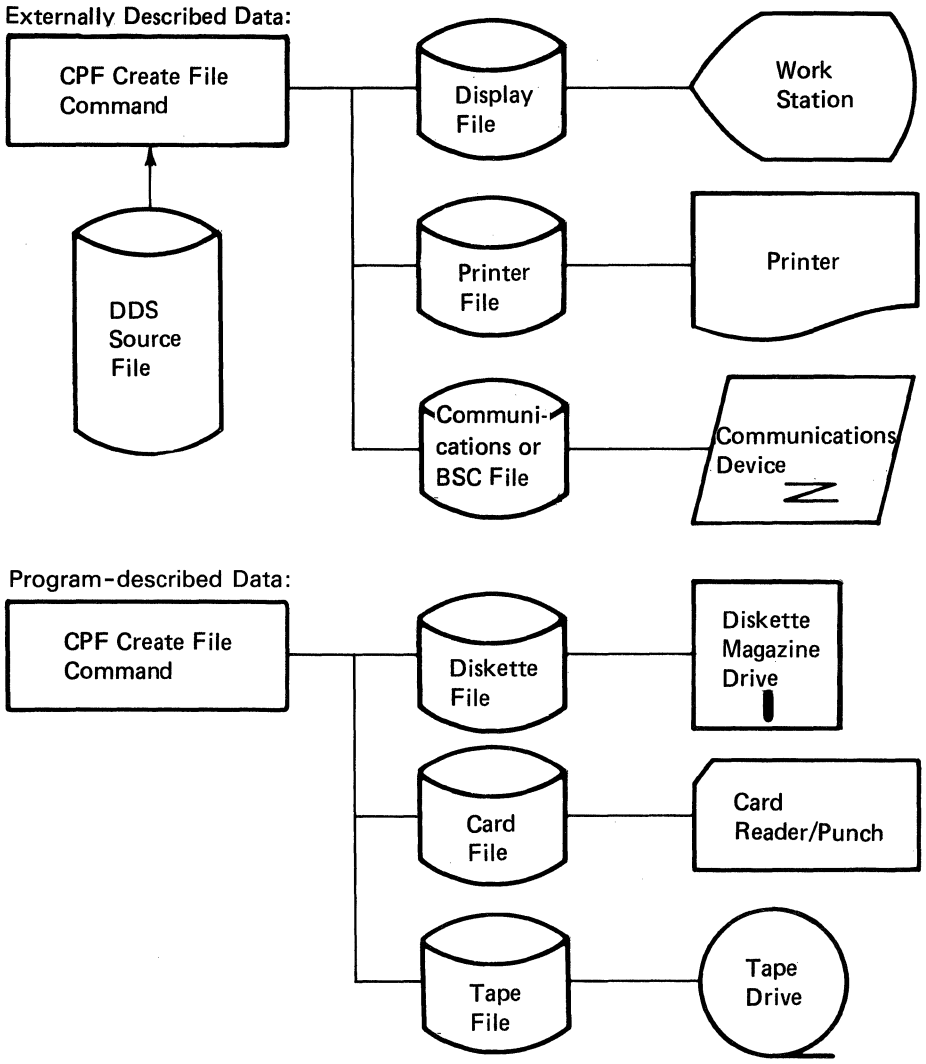


Figure 1-4. Ascending Keyed Sequence, Descending Keyed Sequence, and Arrival Sequence Access Paths. Logical files A and B, with keys specified (key field specifications), determine the order of records written to or read from the physical file. Logical file C, with no key specified, does not.

Device Files

Device files describe how data is processed through input/output devices attached to System/38, such as display devices, printers, and diskettes. Each device used by a program has a device file, which is an executable CPF object created through the use of a create file command. Some of these device files (those used for display devices, printers, and communications or BSC devices) can use DDS to define file, record, and field characteristics. When they do so, they are called *externally described files* because the data is described outside of the programs that use the devices. Other device files (those used for cards, tapes, and diskettes) cannot use DDS and must be program-described device files. For these devices, field-level characteristics must be described in the program itself. (Note that you can have a program-described file that is externally described using DDS. See the appropriate high-level language manual for information on how to do this.)

The following diagrams illustrates the difference between using DDS when creating externally described files and not using DDS when creating program-described files:



Note: Display and printer files can also be program-described.

Coding Characteristics of Device Files

Device files permit more than one record format.

Device files permit the use of option indicators.

Display files and printer files require the locations of fields (except for special cases); communications files and BSC files do not allow them.

Display files, communications files, and BSC files have specifications related to both input and output as follows:

- In display files, you can define a field as input only, output only, output/input, message, or hidden.
- In display files, communications files, and BSC files, indicators can be used for output (option indicators) or input (response indicators).

THREE STEPS TO CREATING AN EXTERNALLY DESCRIBED FILE

The three general steps to creating a file are:

1. Fill in the Data Description Specifications form.
2. Enter the DDS source statements into a source file. The source file can be part of the System/38 data base (in a source physical file such as the IBM-supplied QDDSSRC) or it can be on cards or diskette.
3. Create the file using the appropriate create file command.

This manual is concerned primarily with step 1; however, an overview of all three steps is provided in the following sections.

Note: Through the SDA utility, display files can be created and tested without coding DDS directly. See the *SDA Reference Manual and User's Guide.*

After a physical file has been created, records can be entered into it and grouped by members. This can be done through the use of the data file utility (DFU), which is part of the Interactive Data Base Utilities (IDU) Licensed Program, program 5714-UT1. This utility is described in the *IBM System/38 DFU Reference Manual and User's Guide*, SC21-7714. If you do not have this utility, you can write a high-level language program for data entry.

Filling In the Data Description Specifications Form

A sample Data Description Specifications form is printed in reduced size in Figure 1-5.

The left side of the Data Description Specifications form (positions 1 through 44) is for fixed-format entries called positional entries. Positional entries define the most common attributes of record formats and fields, such as names and lengths of fields. For a brief description of the most important positional entries, see items 1 through 7 following; for details, see the sections on *Positional Entries* in Chapters 2 through 6.

The right side of the Data Description Specifications form (positions 45 through 80) is for DDS keywords. DDS keywords define less-common and more-varied attributes of files, record formats, and fields; they follow a subset of the syntax rules for control language. For a brief description of keywords, see item 8 following; for details, see the sections on *Keyword Entries* in Chapters 2 through 6.

Positional Entries

1. *Sequence Number* and *Form Type* are optional in DDS. *Form Type* identifies the source as DDS source. The entries are valid for all types of files.

2. An asterisk in position 7 makes the entire line a comment (for all types of files). When A (And), O (Or), or blank is in position 7, positions 8 through 16 can provide conditioning for the DDS on or immediately following the current line (conditioning is not valid in physical or logical files).

3. *Name Type* (position 17) identifies the *Name* entry (positions 19 through 28) as a record format name (for all types of files), a field name (for all types of files), a key field name (for physical and logical files only), or a select/omit field (for logical files only).

4. An R specified in position 29 indicates that attributes of the field named in *Name* are to be referenced from a field specified elsewhere (ignored for logical files).

5. *Length*, *Data Type*, and *Decimal Positions* specify attributes of named fields within record formats (valid for all types of files).

6. *Usage* specifies fields as input, output, output/input, hidden, or message fields for device files. Each type of device file has its own restrictions regarding field usage.

7. *Location* specifies the location of the field on the display screen or on the printed page (for display and printer files only).

Keyword Entries

8. *Functions* specified through the use of *keywords* apply at the file, record, or field level (for all types of files), or at the key field level (for physical and logical files only), or at the select/omit field level (for logical files only). Also, constants specified within apostrophes become the default values for displayed or printed fields (for display and printer files only; see the DFT keyword descriptions in the respective files).

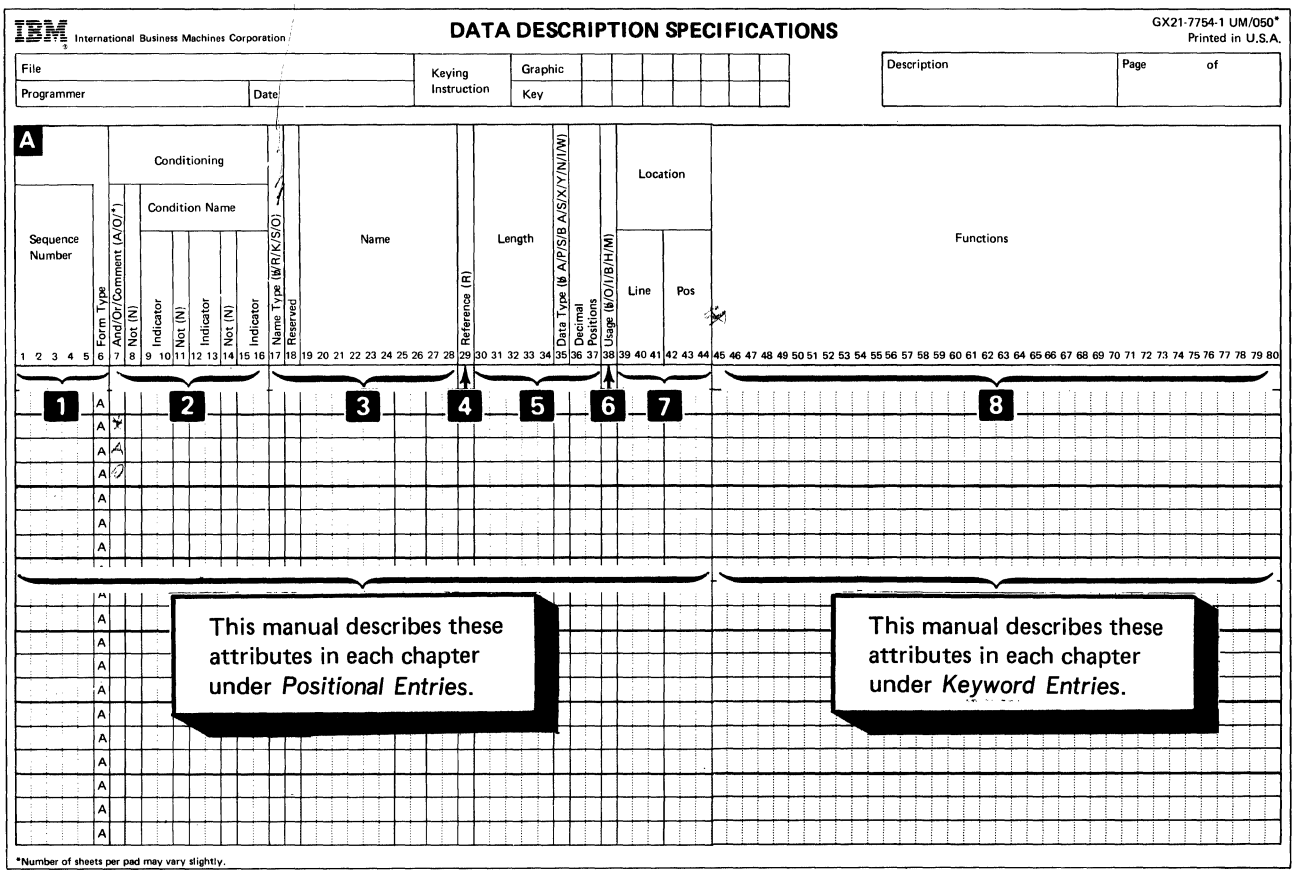


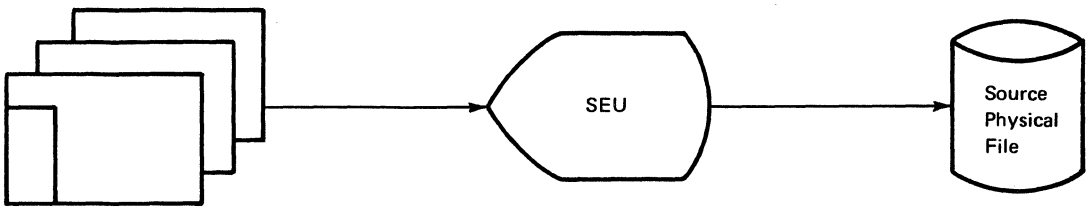
Figure 1-5. Overview of the Positional and Keyword Entries on the Data Description Specifications Form

Entering Data Description Specifications Source Statements

After filling out the Data Description Specifications forms, you must enter the source into source files. You can enter the source interactively or in batch.

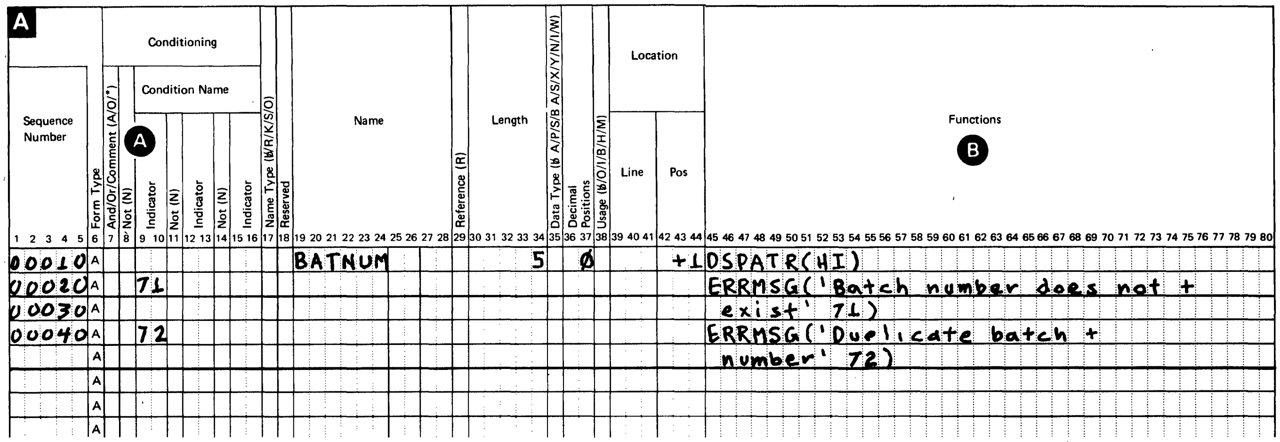
Interactively (Using SEU)

You can use the source entry utility (SEU) of the Interactive Data Base Utilities Licensed Program (IDU):



Enter the Edit Source (EDTSRC) command to call SEU. (See the *Source Entry Utility Reference Manual and User's Guide* for further information on using SEU to enter or update DDS source.)

Sample SEU Display: The following shows how DDS source statements, specified on the Data Description Specifications form, would appear when being entered through SEU. Line 45 is about to be entered. Notice how the option indicator at Ⓐ and the functions (keywords) field at Ⓑ appear on the SEU display.



Sequence Number	Form Type	Indicator (Ⓐ)	Name	Length	Decimal Positions	Functions (Ⓑ)
00010	A		BATNUM	5	Ø	+1DSPATR(HI)
00020	A	71				ERRMSG('Batch number does not +
00030	A					exist' 71)
00040	A	72				ERRMSG('Duplicate batch +
	A					number' 72)

SEU SCREEN

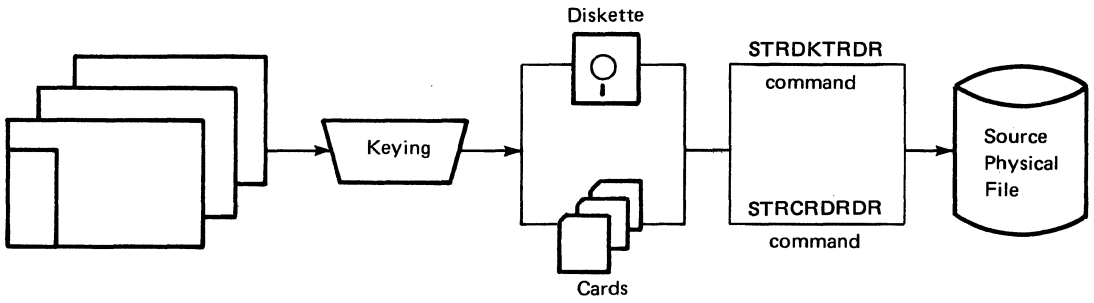
```
SEU     LS W:6      Mbr: MLG105D               Scan: ____________________
FMT DP AAN01N02N03T.Name++++++RLen++TDpBLinPosFunctions++++++++++++++++++++
0042.00 A          BATNUM      5  0      +1DSPATR(HI)
0043.00 A  71                              ERRMSG('Batch number does not +
0044.00 A                                   exist' 71)
.......
        *******END OF DATA*******

FMT SEQNBR   A/O N01N02N03 Name-Type Name      Ref Len   Data-T
DP  _______   _  72______   _        _____      _  _____   _
     Dec Use Line Pos Functions
     ___  _  ___  ___ ERRMSG('Duplicate batch +__________
```

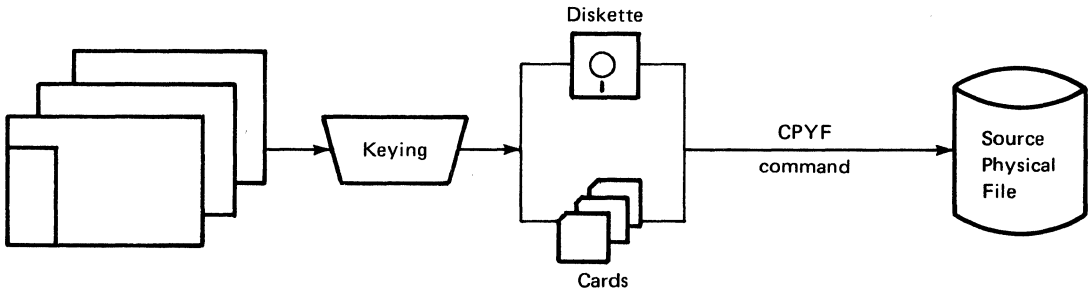
In Batch (Using Cards or Diskette)

You can use one of the following methods:

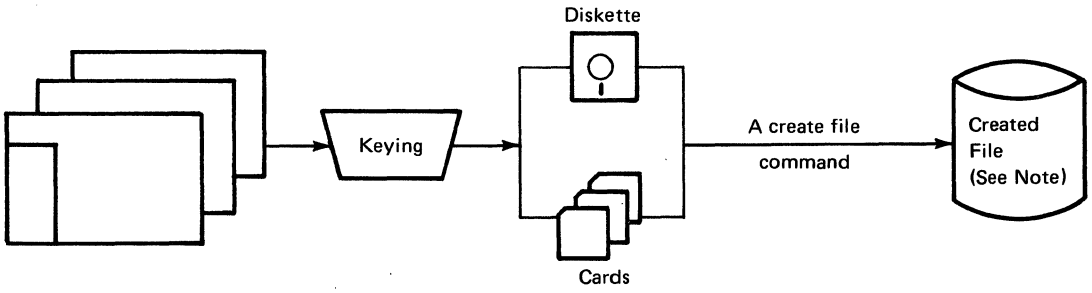
- Enter an input stream containing DDS source and CL commands on cards or diskette and start a spooling reader (STRCRDRDR or STRDKTRDR command):



- Enter only source statements on cards or diskette and copy the resulting data file into a source physical file (CPYF command):



- Enter only source statements on cards or diskette and enter a create file command (specify the name of the data file on the SRCFILE parameter and *FILE on the SRCMBR parameter of the create file command):



Note: When using this method, no source physical file is created.

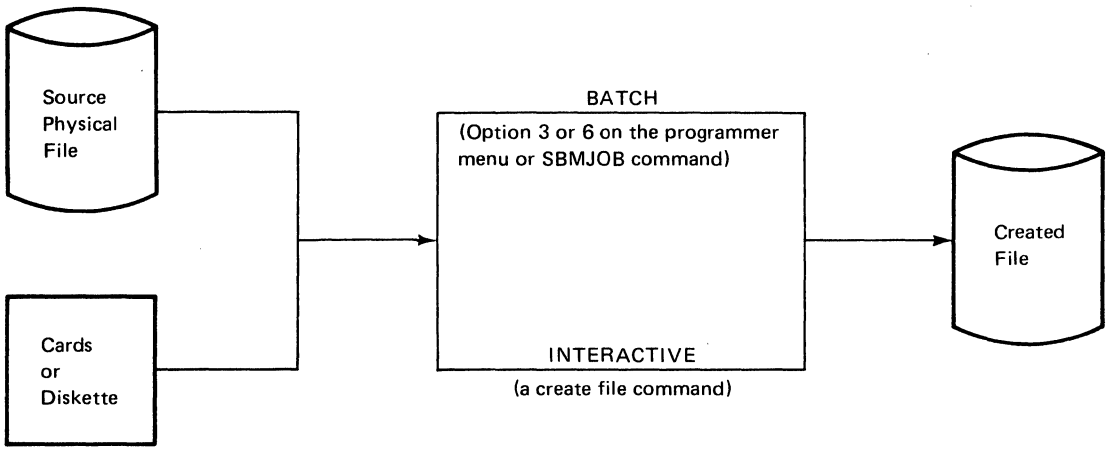
Creating Files

You create files on System/38 by issuing a create file command. The particular create file command to use depends on the type of file you are creating:

Physical file	CRTPF
Logical file	CRTLF
Display file	CRTDSPF
Printer file	CRTPRTF
Communications file	CRTCMNF
BSC file	CRTBSCF

Parameter values on the commands identify some of the attributes of the file, such as the file name, whether the file can be shared, and authority for the file. For card, tape, and diskette files, DDS is not valid.

A create file command invokes the data description processor for externally described files. You can enter the create file command from a work station either interactively or in batch. (See the *Programmer's/User's Work Station Guide*.) The data description processor retrieves the DDS from the source file designated on the create file command, validates the specifications, and creates a data base or device file and a DDS source listing (also called a compiler listing):



The DDS source listing is a printout of the file's data description; any error condition that relates to the data description specifications is identified in this listing. If there are no serious errors in the DDS, the file is created. Appendix F shows a sample DDS source listing and a reduced copy of the Data Description Specifications Debugging Template (used to interpret fields on the DDS source listing).

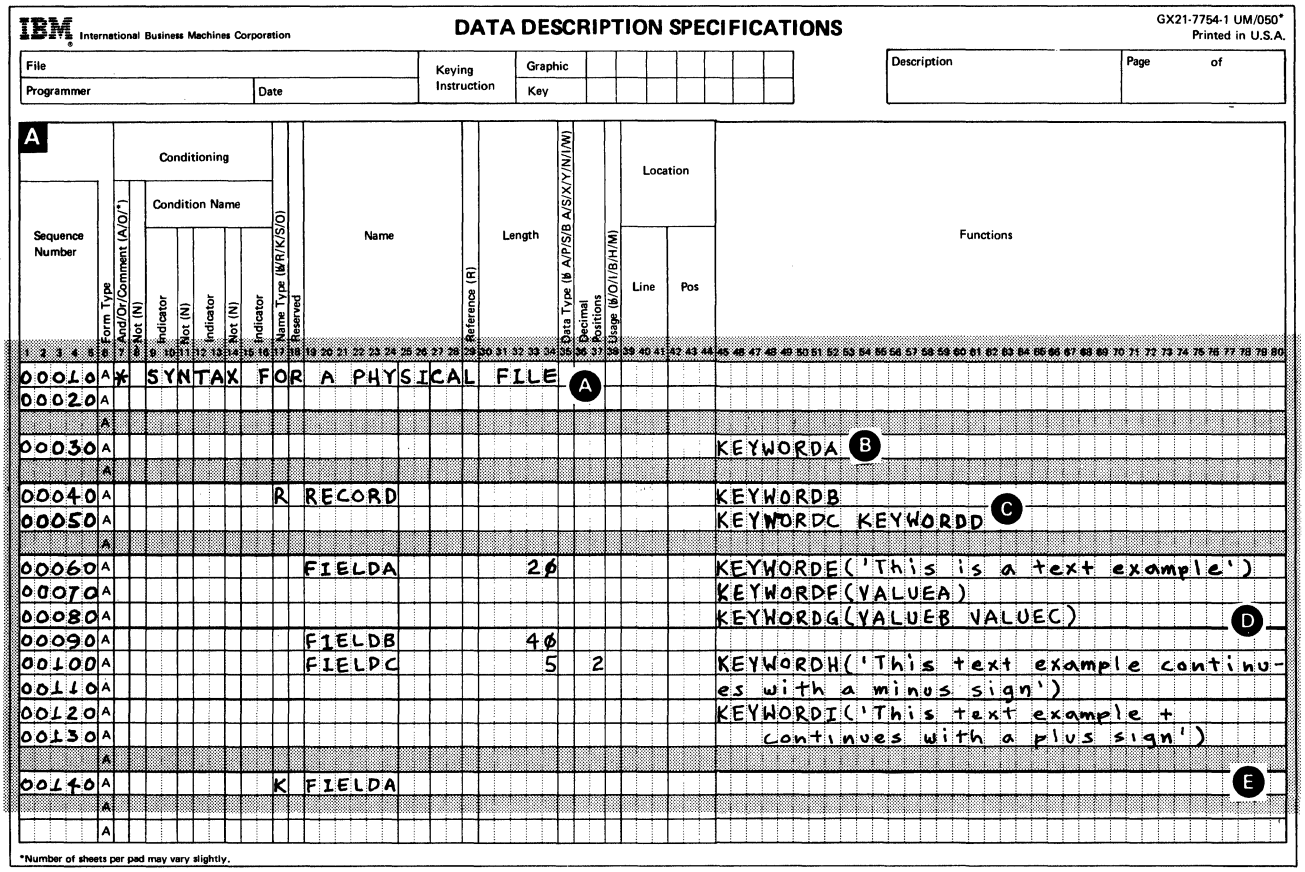
Describing Files Level by Level

When you use data description specifications, you specify the following items, as illustrated in Figures 1-6 through 1-10:

- File level keywords
- Record format names and record level keywords
- Field names, field attributes, and field level keywords
- Key field names and key field level keywords (physical and logical files only)
- Select/omit field names and select/omit level keywords (logical files only)

SYNTAX CODING EXAMPLES

In Figures 1-6 through 1-10, the keywords shown are not actual keywords (except for PFILE in Figure 1-7). They only show where to specify actual keywords.



IBM International Business Machines Corporation — DATA DESCRIPTION SPECIFICATIONS — GX21-7754-1 UM/050* Printed in U.S.A.

File | Keying Instruction: Graphic / Key | Description | Page of

Programmer | Date

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Pos	Functions
00010	A	*		SYNTAX FOR A PHYSICAL FILE															A
00020	A																		
	A																		
00030	A																		KEYWORDA B
	A																		
00040	A								R		RECORD								KEYWORDB
00050	A																		KEYWORDC KEYWORDD C
	A																		
00060	A										FIELDA		20						KEYWORDE('This is a text example')
00070	A																		KEYWORDF(VALUEA)
00080	A																		KEYWORDG(VALUEB VALUEC) D
00090	A										FIELDB		40						
00100	A										FIELDC		5	2					KEYWORDH('This text example continu-
00110	A																		es with a minus sign')
00120	A																		KEYWORDI('This text example +
00130	A																		continues with a plus sign')
	A																		
00140	A								K		FIELDA								E
	A																		
	A																		

*Number of sheets per pad may vary slightly.

A Comments (optional): Comments can appear on any line in DDS. They are identified by an asterisk in position 7.

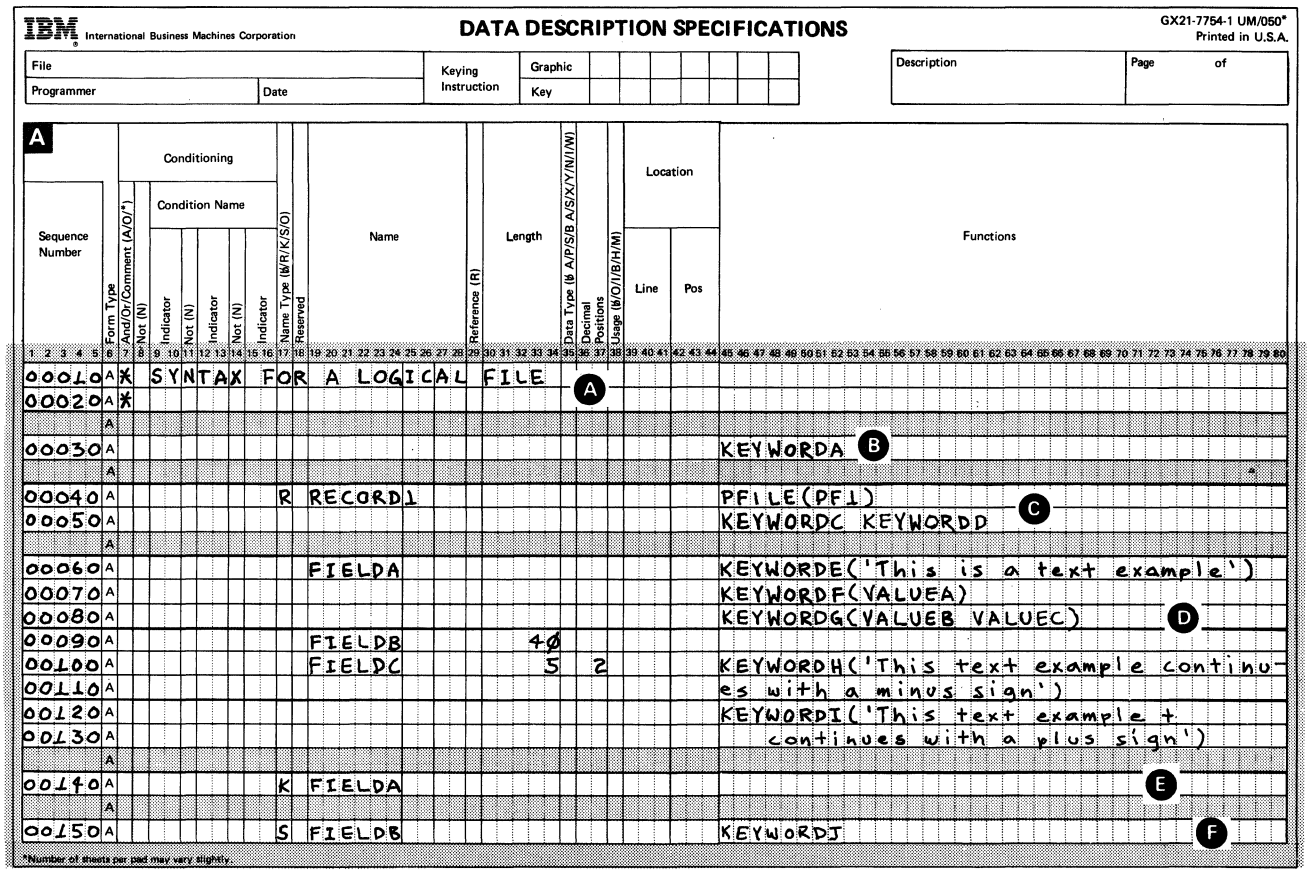
B File level (optional): File level keywords appear before the record format name (RECORD on line 00040).

C Record level (only one allowed in physical files): The R in position 17 identifies RECORD as a record format name. The record level continues until the first field is named.

D Field level (at least one field name required): For fields in physical files, at least a name and length must be specified. Other attributes can be specified explicitly or by default.

E Key field level (optional): The key field level is specified by repeating a field name (here, FIELDA) after the field level specifications. The K specified in position 17 is required at the key field level.

Figure 1-6. Syntax for a Physical File



A Comments (optional): Comments can appear on any line in DDS. They are identified by an asterisk in position 7.

B File level (optional): File level keywords appear before the first record format name (RECORD1 on line 00040).

C Record level (at least one required): The R in position 17 identifies RECORD1 as a record format name. In logical files, the PFILE keyword is required for every record format. The record level continues until the first field is named.

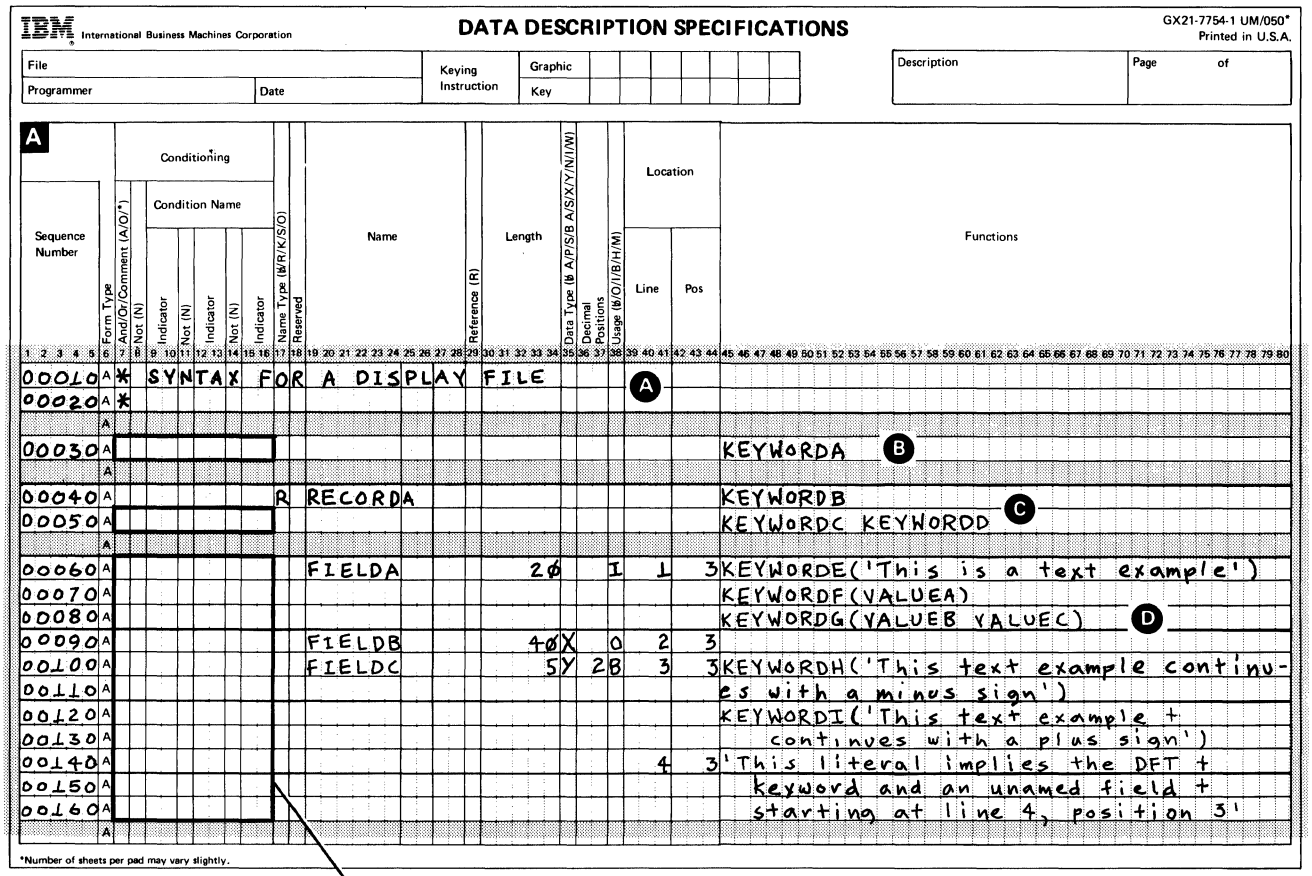
D Field level: Field names and field attributes are not required for logical files. See Chapter 3, *Logical Files*, for a description of the various ways to specify a logical file record format.

E Key field level (optional): The key field level is specified by repeating one or more field names (such as FIELDA) after the field level specifications. The K specified in position 17 is required at the key field level.

F Select/omit level (optional): The S in position 17 identifies FIELDB as a select/omit field (an O could also be specified in position 17 at the select/omit level). The select/omit level follows the key field level.

Note: Items **C** through **F** can be repeated to specify new record formats within the logical file.

Figure 1-7. Syntax for a Logical File



A Comments (optional): Comments can appear on any line in DDS. They are identified by an asterisk in position 7.

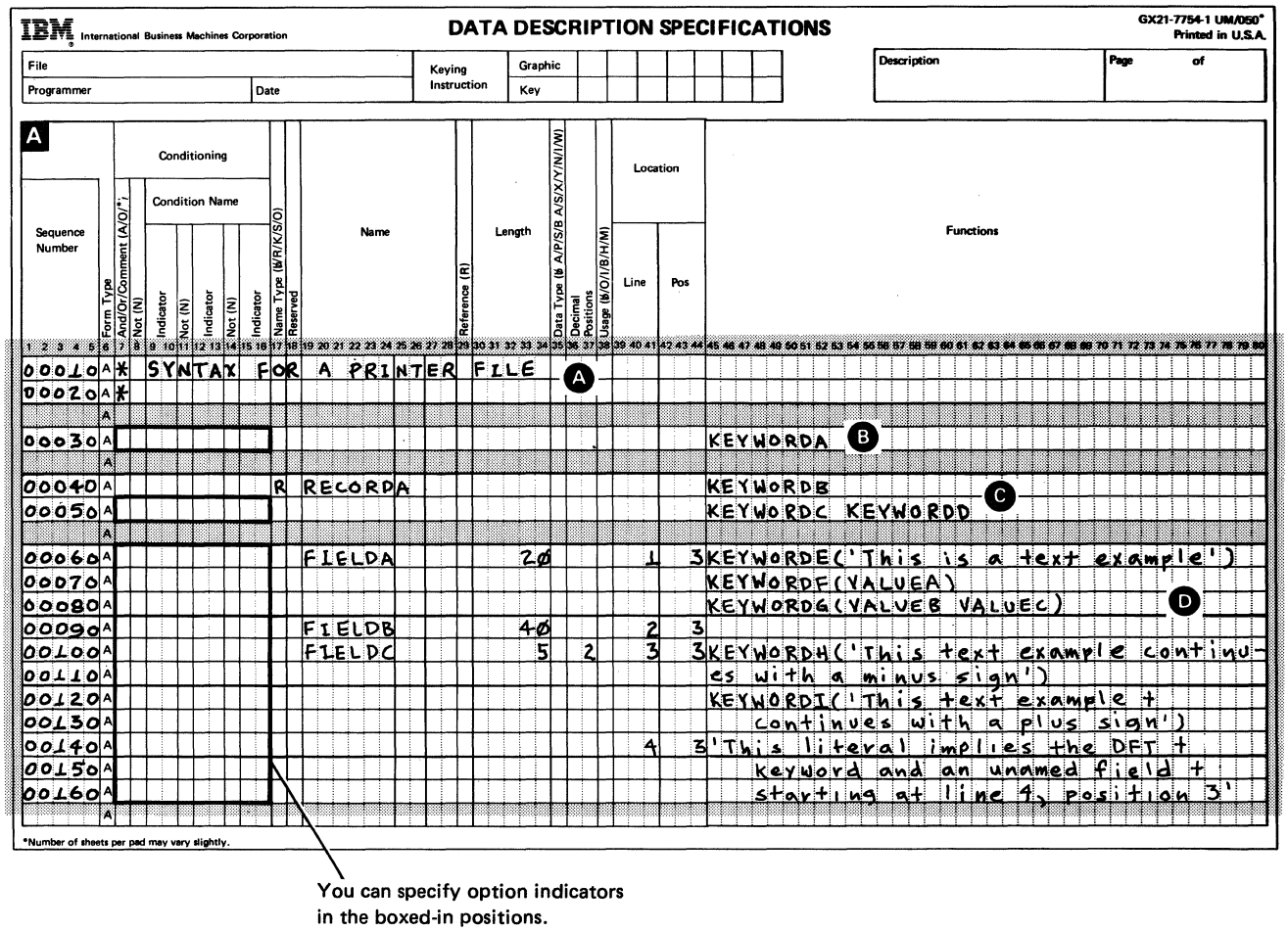
B File level (optional): File level keywords appear before the first record format name (RECORDA on line 00040).

C Record level (at least one required): The R position 17 identifies RECORDA as a record format name. The record level continues until the first field is specified.

D Field level (optional): Display file fields that are to be passed between the display device and your program must be named fields and must have a length specified. Other attributes can be specified explicitly or by default. Constant (unnamed) fields require only a location and a keyword, as described in the DATE, DFT, and TIME keyword descriptions in Chapter 4. Positions 17 through 38 do not apply to constant fields.

Note: Items **C** through **D** can be repeated to specify new record formats within the display file.

Figure 1-8. Syntax for a Display File



A Comments (optional): Comments can appear on any line in DDS. They are identified by an asterisk in position 7.

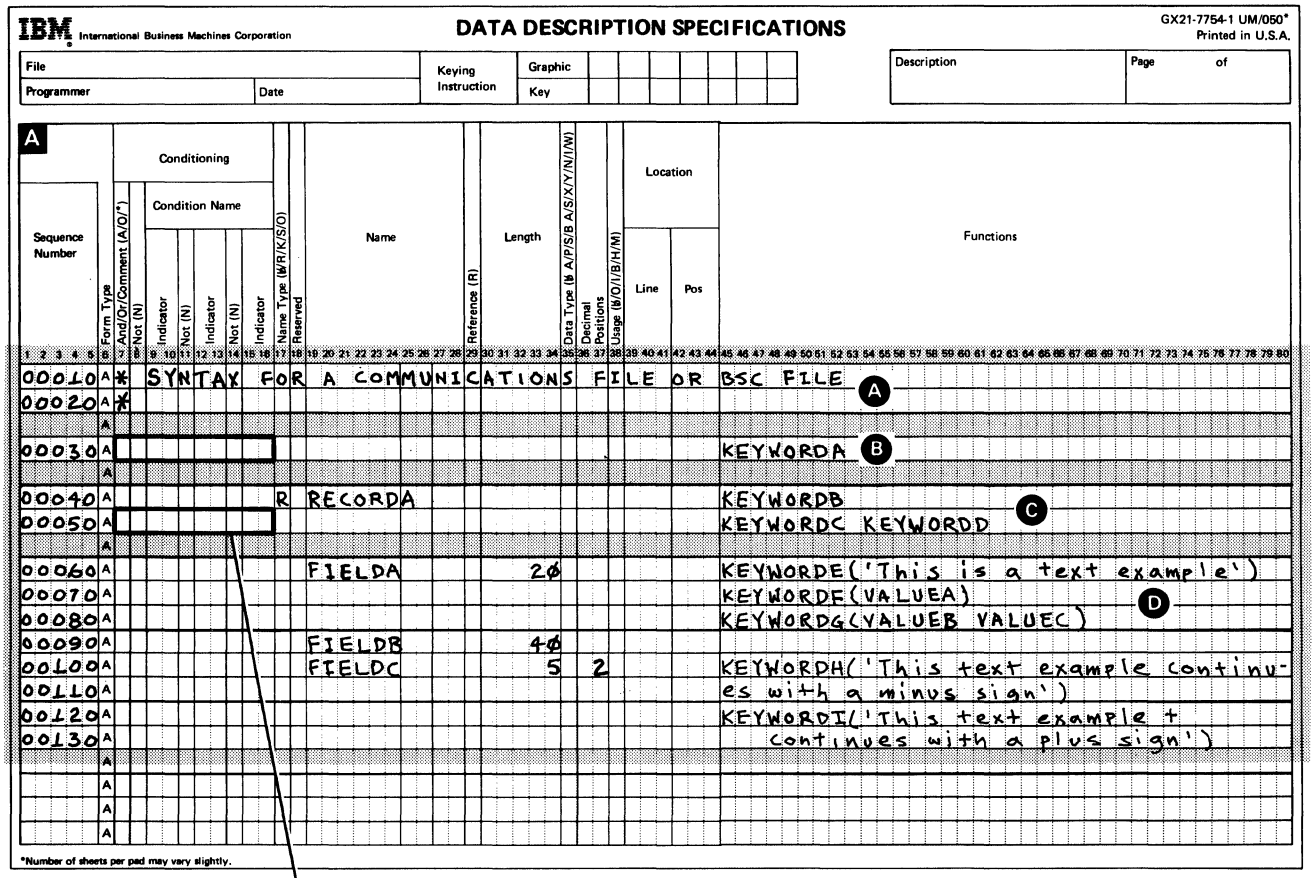
B File level (optional): File level keywords appear before the first record format name (RECORDA on line 00040).

C Record level (at least one required): The R position 17 identifies RECORDA as a record format name. The record level continues until the first field is specified.

D Field level (at least one field, whether named or unnamed, is required in each record format in the file): Printer file fields that are to be passed from your program to the printer must be named fields and must have a length specified. Other attributes can be specified explicitly or by default. Constant (unnamed) fields require only a location and a keyword, as described in the DATE, DFT, PAGNBR, and TIME keyword descriptions in Chapter 5. Positions 17 through 38 do not apply to constant fields.

Note: Items **C** through **D** can be repeated to specify new record formats within the printer file.

Figure 1-9. Syntax for a Printer File



(A) Comments (optional): Comments can appear on any line in DDS. They are identified by an asterisk in position 7.

(B) File level (optional): File level keywords appear before the first record format name (RECORDA on line 00040).

(C) Record level (at least one required): The R position 17 identifies RECORDA as a record format name. The record level continues until the first field is specified.

(D) Field level (at least one field, whether named or unnamed, is required for at least one record format in the file): Communications and BSC file fields must have at least a name (as in FIELDA) and a length. Other attributes can be specified explicitly or by default.

Note: Items (C) through (D) can be repeated to specify new record formats within the communications or BSC file.

Figure 1-10. Syntax for a Communications File or a BSC File

DATA DESCRIPTION SPECIFICATIONS SYNTAX RULES

The syntax for coding DDS keywords and their parameter values is a compatible subset of the syntax used for the control language. Figures 1-6 through 1-10 show how to use this syntax for each type of file. The general rules are:

- Code all DDS entries in all uppercase except for character literals.
- Code keywords on the same line as the entry with which they are associated, or on subsequent lines.
- Separate multiple keywords with one or more blanks. Parameter values for keywords must be enclosed in parentheses with the initial parenthesis immediately following the keyword, for example,

  KEYWORD(VALUE)

  (This rule differs slightly from control language because, when coding control language, parameter values can be positional. Syntax for data description specifications requires that the keyword be specified, except when specifying a constant or the parameter value for the DFT keyword.)
- Separate multiple parameter values for the same keyword with one or more blanks. For example,

  KEYWORD(VALUEA VALUEB)
- Use a period to separate the components of a qualified name. Embedded blanks are not allowed. For example,

  KEYWORD(file.library)
- Use apostrophes to enclose character literals. (Numeric literals appear without apostrophes; see the coding examples for the COMP, RANGE, and VALUES keywords.) Character literals can appear in two places in the syntax for DDS:
  - As a parameter value for some keywords. For example, TEXT (in all types of files) and COLHDG (in data base files) require character literals as text description. Other keywords such as CAnn and CFnn use character literals as text descriptions for response indicators.
  - With or without the DFT keyword, for display and printer files only, as the default value of a constant field. In printer files a character literal can also be specified for named fields. Even if you do not specify the DFT keyword, specifying a character literal implies the DFT keyword.

- Use double apostrophes ('') when specifying any apostrophes within a quoted parameter value so that one apostrophe will appear in output. For example,

  KEYWORD('Customer''s name')

  appears as

  Customer's name

- Use a plus (+) or minus (-) sign to indicate that a keyword and its parameter values are continued on the succeeding line. The sign (+ or -) that you choose for a continuation character has a particular meaning as follows:
  - A *minus (-) sign* indicates that the continuation is to begin in position 45 of the next line (the first position of the functions field).
  - A *plus (+) sign* indicates that the continuation is to begin with the first nonblank (first significant) character in the functions field on the next line.

  If you specify a continuation character within a parameter value (such as a character literal), any blanks preceding the continuation character are included in the parameter value.

  The sign must be the last nonblank character in the functions field. One of the signs must be used when the entire specification for a keyword does not fit on a single line. A single statement can be continued to a maximum of 2000 character positions.

- Specify a plus (+) sign as the last nonblank character on a line to continue conditioning for keywords specified on the next line. This is especially helpful when a condition includes several option indicators and applies to several keywords.

- CPF implicitly continues a DDS statement until you specify one of the following:
  - A record format name (R in position 17)
  - A field specification (field name or location)
  - For device files, an option indicator or condition name that conditions a field or a field location

- The keyword descriptions use the following punctuation marks:

  [ ] Enclosed values are optional.

  [...] Specify additional values as needed.

  { } The upper value is the default value (see REFFLD).

  | Specify either the value to the left or the value to the right (may refer to optional values).

Chapter 2. Physical Files

This chapter is divided into two sections. The first, *Positional Entries*, provides rules and examples for filling in positions 1 through 44 of the Data Description Specifications form. To find information in this section, first determine what position on the form to use, then look up the section describing that position.

The second section of this chapter, *Keyword Entries*, provides rules and examples for specifying DDS keywords. The keywords are described in alphabetical order.

For guidance in choosing positional entries and keywords for physical files, see the *CPF Programmer's Guide*.

Note: If you are using DDS to describe a source file, you cannot use the ABSVAL, DESCEND, NOALTSEQ, SIGNED, UNIQUE, or ZONE keywords.

DEFINING A PHYSICAL FILE

A physical file can contain only one record format. Specify the record format in either of two ways:

- *Define a new record format*: Specify field and key field specifications as desired for the new record format.
- *Share an existing record format*: Specify, through the FORMAT keyword, that CPF is to use a previously defined record format from a physical or a logical file. When the FORMAT keyword is used, key field level specifications must be specified again even if they were specified on the existing record format.

To define a physical file, specify the entries in the following order:

1. File level entries (optional)
2. Record level entries
3. Field level entries
4. Key field level entries (optional)

Note: The file name is provided through the Create Physical File (CRTPF) command, not through DDS.

Complete physical file examples can be found in Appendix F, *Examples*.

An explanation of *file level*, *record level*, *field level*, and *key field level* can be found in Chapter 1, *Introduction*.

The syntax rules for specifying DDS keywords can be found in Chapter 1, *Introduction*.

A physical file cannot have more than one record format. The maximum number of fields in the record format is 8000; the maximum number of bytes in the record format is 32 766.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/060* Printed in U.S.A.

File		Keying Instruction	Graphic								Description	Page	of
Programmer	Date		Key										

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (I/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (I/ A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (I/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A	*									PHYSICAL FILE CODING EXAMPLE								
00020	A																		REF(INVENTORY.INVENCTL)
00030	A																		UNIQUE
00040	A								R		ORDFMT								TEXT('Format For Purchase Orders')
00050	A										ORDNBR		7		Ø				COLHDG('Order' 'Number')
00060	A										ITMNBR	R	1Ø						
00070	A										SUPNBR	R	+2						REFFLD(SUPID SUPMST.SUPLIB)
00080	A										QTYORD		5B						
00090	A								K		ORDNBR								
00100	A								K		ITMNBR								ABSVAL
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

*Number of sheets per pad may vary slightly.

Figure 2-1. Physical File Coding Example

Positional Entries (Positions 1-44)

This section describes how to specify the first 44 positions of the Data Description Specifications form for physical files. To code the rest of the form, see the section *Keyword Entries (Functions Field, Positions 45-80)* later in this chapter.

Figure 2-1 shows some positional entries for physical files.

Positions 1-5 (Sequence Number)

Use these positions to specify a sequence number for each line on the form. The sequence number is optional and is used for documentation purposes only.

Position 6 (Form Type)

The A in this position designates this form as a Data Description Specifications form. The form type is optional and is for documentation purposes only.

Position 7 (Comment)

An asterisk (*) in position 7 identifies the line as a comment line. Comment lines can appear anywhere in DDS and are retained only in the source file and printed on the source listing. (Comments are not printed on the expanded source listing.) Use positions 8 through 80 for comment text. A blank line (no characters specified in positions 7 through 80) is treated as a comment line.

Positions 8-16 (Conditioning)

These positions do not apply to physical files and must be blank unless an asterisk appears in position 7 (which indicates a comment line).

Position 17 (Name Type)

The value in this position identifies the type of name in positions 19 through 28 (name). The valid entries are:

Entry	Meaning
R	Record format name
Blank	Field name
K	Key field name

Note: There can be only one R specified for a physical file because a physical file can contain only one record format.

Position 18 (Reserved)

This position does not apply to any file and must be blank unless an asterisk appears in position 7 (indicates a comment line).

Positions 19-28 (Name)

Use these positions to specify the names of the following:

- The record format for this physical file
- The field or fields within the record format (unless you specify the FORMAT keyword on the record format name)
- The field or fields to be used as key fields (to establish a keyed sequence access path)

Note: The file name is provided through the create file command, not in the DDS.

When specifying names in DDS, the following rules apply:

- Names must be 10 characters or less.
- Names must start in position 19.
- A name must begin with an alphabetic character (A through Z, @, $, and #). All subsequent characters can be alphameric (A through Z, 0 through 9, @, $, #, and underscore (_)). There can be no embedded blanks.

Figure 2-1 shows how to code the name field; it shows a record format name (R specified in position 17), a field name (blank specified in position 17) and a key field name (K specified in position 17).

High-level languages can impose specific length and value restrictions on the name. For example, RPG III accepts only field names of 6 characters or less and record format names of up to 8 characters, and they cannot contain an underscore. It is your responsibility to ensure that the name syntax used is acceptable to all language processors that use the file.

Record Format Name

When you specify R in position 17 (name type), the name specified in positions 19 through 28 is a record format name. You can specify only one record format name for a physical file. Specify the record format name in either of two ways:

- As the name of a *new* record format with field names specified in this physical file. The name of the record format can be the same as the file name specified in the create file command; however, a warning message is issued if the names are not unique, because some language processors do not allow record format and file names to be the same. RPG III is such a language.

- As the name of a record format *previously described* in a physical or a logical file. Field names and attributes are not specified and the FORMAT keyword must be specified. The FORMAT keyword is explained in the *Keyword Entries* section in this chapter.

Field Name

When position 17 (name type) is left blank, the name specified in positions 19 through 28 is a field name. Field entries describe the characteristics of a field of data within the record. Field names must be unique within the record format. The order in which you specify the field names is the order in which the fields appear within the physical record. In physical files, all fields must be named.

Key Field Name

When you specify K in position 17 (name type), the name specified in positions 19 through 28 is a key field name. It must be one of the field names within the physical file record format. The contents of this field are used to sequence the records for retrieval from the data base. Specifying a key is optional; the default is to sequence the records in arrival sequence (the sequence in which they were put into the data base). See the *CPF Programmer's Guide* for general information on access paths and for information on the effect of keyed or arrival sequence access paths on save/restore operations.

A key can have more than one key field; such a key is a *composite key*. For a composite key, you must specify the field names in the order of importance (major to minor), with each field name on a separate specification line. The number of fields that make up a key is restricted; the total key length must not exceed 120 bytes.

For example, suppose a key consists of fields named FIELDA, FIELDB, and FIELDC (in that order). The DDS appears as follows:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A	*									SAMPLE COMPOSITE KEY (PHYSICAL FILE)								
00020	A								R		RECORD1								
00030	A										FIELDA		3		0				
00040	A										FIELDB		3		0				
00050	A										FIELDC		3		0				
00060	A										FIELDD		3		0				
00070	A								K		FIELDA								
00080	A								K		FIELDB								
00090	A								K		FIELDC								
	A																		
	A																		
	A																		

These three fields make up the composite key

Upon retrieval, the records are sequenced according to the contents of FIELDA; if two or more records with the same value in FIELDA exist, CPF sequences those records according to the values in FIELDB. If there are two or more of those records with the same value in both FIELDA and FIELDB, they are sequenced according to the values in FIELDC.

Consider the following file:

Record	FIELDA	FIELDB	FIELDC
1	333	99	67
2	444	10	45
3	222	34	23
4	222	12	01
5	222	23	45
6	111	06	89
7	222	23	67

Assuming ascending sequencing for all fields, the records are retrieved in this order:

Record	FIELDA	FIELDB	FIELDC
6	111	06	89
4	222	12	01
5	222	23	45
7	222	23	67
3	222	34	23
1	333	99	67
2	444	10	45

Note that records 3, 4, 5, and 7 have the same contents in FIELDA, so FIELDB becomes the determining field. Within those four records, 5 and 7 have the same values in FIELDB, so for these two records, FIELDC becomes the determining field. If FIELDC also contains duplicate values, the records are retrieved in first-in-first-out (FIFO) order. FIFO is the default order. Last-in-first-out (LIFO) processing is available to change this order, and the UNIQUE keyword can be specified to prevent duplicate key values. LIFO and UNIQUE are described in *Keyword Entries* later in this chapter.

See the SIGNED keyword description for an example that includes a key field with negative (-) contents.

There are special restrictions that apply to the specification of key fields when FILETYPE(*SRC) is specified on the Create Physical File (CRTPF) or the Create Source Physical File (CRTSRCPF) command. See the *CPF Programmer's Guide* for information about key field specification for source files.

Position 29 (Reference)

Specify R in this position to use the reference function of CPF to copy the attributes of a previously defined named field (called the *referenced field*). For physical files, you must also specify the REF or the REFFLD keyword. The referenced field can be previously defined in the physical file you are defining; or it can be in a previously created data base file. The field attributes referenced are the field's length, data type, and decimal positions, as well as editing, validity checking, column heading, and text keywords.

If you do not specify R, you cannot use the reference function for this field and you must specify field attributes for this field.

Position 29 must be blank at the file and record levels.

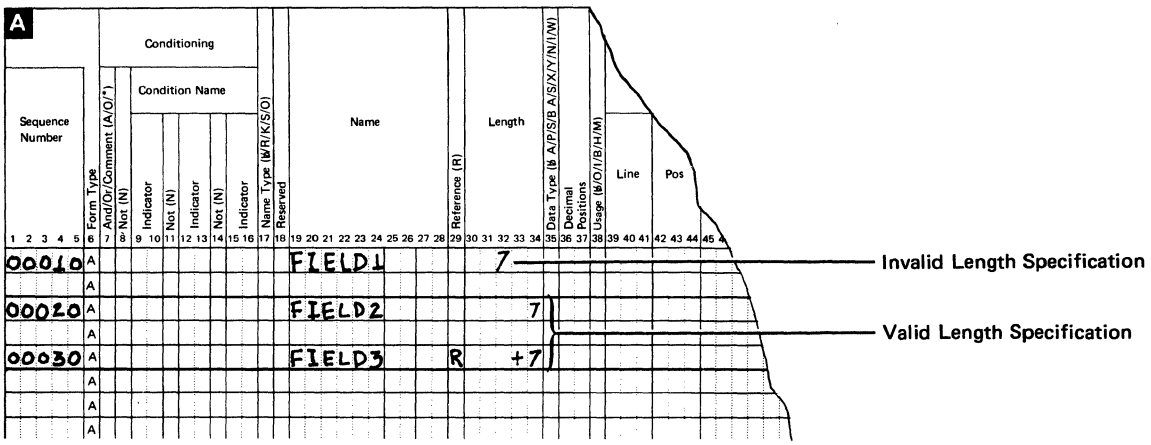
The name of the referenced field cannot be the same as the field you are defining if the referenced field is in the file you are defining. If the names are the same, you must specify the name of the file defining the referenced field as a parameter value with the REF or REFFLD keyword. If the names are different, you must specify the name of the referenced field with the REFFLD keyword. See the REF and REFFLD keyword descriptions later in this chapter and Appendix A, *How to Specify REF and REFFLD*.

To override specific attributes of the referenced field, specify those attributes for the field you are defining. For example, if you specify the EDTCDE keyword or the EDTWRD keyword on the field you are defining, no editing specifications are copied from the referenced field. Also, if you specify CHECK (AB; ME; MF; M10; M11; VN), COMP, RANGE, or VALUES on the field you are defining, no validity checking specifications are copied from the referenced field. If you specify data type, field length, or decimal positions for the field you are defining, then neither editing nor validity checking keywords are copied from the referenced field.

Note: Once the physical file is created, the referenced file can be deleted or changed without affecting the field descriptions in the physical file. To incorporate changes made in the referenced file, delete and re-create the physical file.

Positions 30-34 (Length)

You must specify a field length unless the length is being duplicated from a referenced field. Specify the number of digits for a numeric type field; specify the number of characters for a character type field. If you specify length, it must be right-justified; leading zeros are optional. The following example shows valid and invalid field length specifications:



Valid length specifications are:

Data Type	Valid Lengths
Character	1 through 32 766 characters
Binary	1 through 9 digits
Zoned decimal	1 through 31 digits
Packed decimal	1 through 31 digits

The sum of the number of bytes occupied by all fields in a record must not exceed 32 766. The number of bytes actually occupied is determined by the system as follows:

Data Type	Bytes Occupied
Character	Number of characters
Binary	
1-4 digits	2 bytes
5-9 digits	4 bytes
Zoned decimal	Number of digits
Packed decimal	(Number of digits/2) + 1

Note: System/38 performs arithmetic operations more efficiently for packed decimal than for zoned decimal data type.

If you are using a referenced field, you can override the length of the field by specifying a new value or by specifying an increase or decrease in length. To increase the length, specify +n, where n is the increase. To decrease the length, specify -n, where n is the decrease. For example, an entry of +4 for a numeric field indicates that it is to be 4 digits longer than the referenced field. Figure 2-1 shows how to change and override the field length.

Note: High-level languages can impose specific length restrictions on the field length; these restrictions should be observed for files used by those languages.

Position 35 (Data Type)

Use this position to specify the data type of the field within the data base. The valid data type entries for physical files are:

Entry	Meaning
Numeric types	
P	Packed decimal
S	Zoned decimal
B	Binary
Alphabetic type	
A	Character

Figure 2-1 shows how to code the data type.

If you do not specify a data type and do not duplicate one from a referenced field, CPF assigns a default value as follows:

- A (character) if the decimal positions (36 through 37) are blank
- P (packed decimal) if the decimal positions (36 through 37) contain a number in the range 0 through 31

Note: Placing 0 in position 37 is a convenient way to specify an integer numeric field for any of the three numeric types.

Positions 36-37 (Decimal Positions)

Use these positions to specify the decimal placement within a packed decimal, zoned decimal, or binary field. Specify a decimal number from 0 through 31 to indicate the number of decimal positions to the right of the decimal point. (The number here must not be greater than the number of digits specified in the field length.) Figure 2-1 shows how to code the decimal positions field.

For all data types, data is actually stored in the system without a decimal point; the decimal point is only implied. For example, the value stored for 1.23 is 123, and this is what appears in display or printer files if editing is not specified.

If you are using a referenced field, these positions can be either overridden or changed. To override the positions, specify the new value explicitly. To change the positions, specify the amount you want the field increased or decreased and precede it with either a + or -, respectively. For example, an entry of +4 indicates there are to be 4 more digits to the right of the decimal point than were in the referenced field. If the resulting number of decimal positions is greater than the maximum allowed, an error message is sent.

Note: High-level languages can impose specific length and value restrictions on the decimal positions; these restrictions should be observed for files used by those languages.

Position 38 (Usage)

This position does not apply to physical files and must be blank unless an asterisk appears in position 7 (which indicates a comment line).

Positions 39-44 (Location)

These positions do not apply to physical files and must be blank unless an asterisk appears in position 7 (which indicates a comment line).

Keyword Entries (Positions 45-80)

This section contains keyword entries valid for defining physical files. They are entered in the functions field (positions 45 through 80). See the section *Data Description Specifications Syntax Rules* in Chapter 1 for a discussion of the general rules for specifying keywords. Figure 2-2 shows how to use the general syntax rules for specifying DDS for physical files.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050*
Printed in U.S.A.

File		Keying Instruction	Graphic								Description	Page of
Programmer	Date		Key									

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/Y/N/I/M)	Decimal Positions	Usage (blank/I/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A	X		SYNTAX FOR A PHYSICAL FILE															
00020	A	X																	
	A																		
00030	A																		KEYWORDA
	A																		
00040	A								R		RECORD								KEYWORDB
00050	A																		KEYWORDC KEYWORDD
	A																		
00060	A										FIELDA		20						KEYWORDE('This is a text example')
00070	A																		KEYWORDF(VALUEA)
00080	A																		KEYWORDG(VALUEB VALUEC)
00090	A										FIELDB		40						
00100	A										FIELDC		5		2				KEYWORDH('This text example continu-
00110	A																		es with a minus sign')
00120	A																		KEYWORDI('This text example +
00130	A																		continues with a plus sign')
	A																		
	A																		
	A																		
	A																		

*Number of sheets per pad may vary slightly.

Figure 2-2. Syntax for a Physical File

The following keywords are valid for physical files:

ABSVAL
ALTSEQ
CHECK
CMP
COLHDG
COMP
DESCEND
DIGIT
EDTCDE
EDTWRD
FORMAT
LIFO
NOALTSEQ
RANGE
REF
REFFLD
SIGNED
TEXT
UNIQUE
VALUES
ZONE

Note: When you use DDS to describe a source file (usually created without DDS, using the CRTSRCPF command), you cannot use the following keywords:

ABSVAL
ALTSEQ
DESCENT
LIFO
NOALTSEQ
SIGNED
UNIQUE
ZONE

ABSVAL

Use this key field level keyword (absolute values) to specify that when CPF sequences the values associated with this numeric key field, it is to ignore the sign of the field.

The following example shows six records with zoned decimal key fields:

Record	Numeric Key Field (Zoned Decimal)	Hexadecimal Representation
1	98	F9F8
2	00	F0F0
3	98-	F9D8
4	97	F9F7
5	20	F2F0
6	99	F9F9

If no sequencing keywords are specified, the records are sequenced in this order:

Record	Numeric Key Field (Zoned Decimal)	Hexadecimal Representation
2	00	F0F0
5	20	F2F0
3	98-	F9D8
4	97	F9F7
1	98	F9F8
6	99	F9F9

If the ABSVAL keyword is specified, the absolute value of the negative field is used, and the resulting sequence is:

Record	Numeric Key Field (Zoned Decimal)	Hexadecimal Representation
2	00	F0F0
5	20	F2F0
4	97	F9F7
1	98	F9F8
3	98-	F9D8
6	99	F9F9

The ABSVAL keyword is not valid for a field of character data type. You can not use this keyword with the SIGNED, the ZONE, or the DIGIT keyword.

ABSVAL (a key field level keyword) and ALTSEQ (a file level keyword) are mutually exclusive; that is, if you specify ABSVAL for a key field, NOALTSEQ is in effect for that key field, even if ALTSEQ was specified at the file level (whether the NOALTSEQ keyword is specified or not).

If you do not specify SIGNED, ABSVAL, ZONE, or DIGIT for a key field, the value of the field is treated as a string of unsigned binary data.

The following is an example of how to specify the ABSVAL keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A										ORDNBR		5		0				
00020	A								K		ORDNBR								ABSVAL
	A																		
	A																		
	A																		

ALTSEQ(alternate-sequence-table-name[.library-name])

This file level keyword (alternate sequence) tells CPF that if a key has been specified for this file, CPF is to use an alternate collating table when sequencing the records of a file member for retreival. Such a table can be created by the Create Table (CRTTBL) command. The name of the alternate collating table is a required parameter value; the library name is optional. (If a library is not specified, the library list, *LIBL, is used.)

The ALTSEQ keyword is not valid when FILETYPE(*SRC) is specified on the Create File command. See the *CPF Programmer's Guide* for information about the FILETYPE(*SRC) parameter value.

Notes:

1. Alternate collating sequences are not valid for key fields with a data type of packed decimal or binary, or for key fields with SIGNED or ABSVAL specified. For these fields, NOALTSEQ (a key field level keyword) is assumed and need not be specified. NOALTSEQ can also be specified for any field in a composite key not requiring the alternate sequence.
2. At file creation time, you must have operational rights to the alternate collating table. The alternate collating table is created using the Create Table (CRTTBL) command. See the *CPF Programmer's Guide* for more information on how to create and enter a table.

In the following example, records with format DISTR are sequenced by key NAME, by means of the alternate collating sequence TABLE1 in library TABLELIB:

A

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Conditioning (8–16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19–28)	Reference (R) (29)	Length (30–34)	Data Type (35)	Decimal Positions (36–37)	Usage (38)	Location Line (39–41)	Location Pos (42–44)	Functions (45–80)
00020	A													ALTSEQ(TABLE1.TABLELIB)
00030	A			R		DISTR								
00040	A					⋮								
00050	A					⋮								
00060	A					⋮								
00070	A					NAME		20						
00080	A					⋮								
00090	A					⋮								
00100	A			K		NAME								
	A													
	A													

CHECK(edit-check-code [. . .])

This field level keyword checks the validity of data entered in a field of a display file by applying one or more edit/check algorithms against the data. The keyword does not request any action by data base data management, but can be specified in the data base file description so that it can be duplicated into display file descriptions when this field is referenced at display file creation. The rules for specifying this keyword are the same for a data base record format as for a display file record format. See Chapter 4, *Display Files*, for information on how to specify this keyword.

The following edit/check codes are not allowed in data base files even for reference purposes:

Codes	Meanings
FE	Field exit check
RL	Right-to-left cursor movement
RLTB	Right-to-left, top-to-bottom cursor movement

CMP(relational-operator-code constant-value)

The CMP keyword is equivalent to the COMP keyword. The COMP keyword is preferred because it is compatible with DDS on other systems. See the COMP keyword description for an explanation of how to use these keywords.

COLHDG('line-1' ['line-2' ['line-3']])

This field level keyword (column heading) specifies column headings to be used as a label for this field by the query utility, the data file utility (DFU), and the screen design aid (SDA). A maximum of three lines of 20 characters each is allowed.

Each line of the column heading must be enclosed in apostrophes. Use double apostrophes ('') to specify apostrophes within column headings. One or more blanks must be used to separate the first column heading line from the second and the second from the third.

If COLHDG is not specified and is not retrieved from a referenced field, the field name is used. If COLHDG is specified and TEXT is not specified, 50 positions of column heading information are used as text. For example, a specification of COLHDG('Order' 'Date') is equivalent to TEXT('Order Date').

The following example shows how to specify the COLHDG keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10 11	12 13	14 15 16			17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00150	A										ORDDAT		5						COLHDG('Order' 'Date')
00160	A										NAME		20						COLHDG('Customer''s Name')
00170	A										CITY		20						COLHDG('Customer' 'City' 'Field')

The following display illustrates how the column headings could appear when executing query, DFU, or SDA.

```
Order                                  Customer
Date          Customer's Name          City
                                       Field

NNNNN         XXXXXXXXXXXXXXXXXXX      XXXXXXXXXXXXXXXXXXX
```

COMP(relational-operator-code value)

This field level keyword (comparison) is a validity checking keyword used when fields are referenced by a display file. It does not request any action by data base data management, but can be specified in the data base file description so that it can be duplicated into device file descriptions when this field is referenced at device file creation. The rules for specifying this keyword are the same for a data base record format as for a device file record format. See Chapter 4, *Display Files*, for information on how to specify this keyword.

DESCEND

Use this key field level keyword to specify that the values of this character or numeric key field are to be retrieved in descending sequence. If this keyword is not specified, ascending sequence is used. See the SIGNED keyword description for an example of data sorted with the DESCEND keyword. The following example shows how to specify the DESCEND keyword:

A

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9–10)	Not (N) (11)	Indicator (12–13)	Not (N) (14)	Indicator (15–16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19–28)	Reference (R) (29)	Length (30–34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36–37)	Usage (b/O/I/B/H/M) (38)	Location Line (39–41)	Location Pos (42–44)	Functions (45–80)
00100	A								K		BALDUE								DESCEND
	A																		

DIGIT

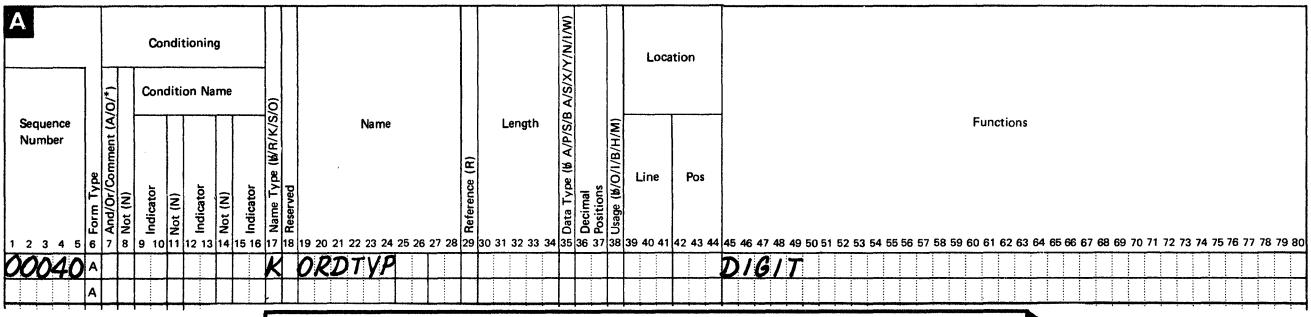
Use this key field level keyword to specify that only the digit portion (rightmost 4 bits) of each byte of the key field is to be used when constructing a value associated with this key field. The zone portion is filled with zeros.

This keyword is applied against the entire key field (not just a position within the field); it is valid only for character or zoned decimal type fields.

You cannot use this keyword with the ABSVAL, the SIGNED, or the ZONE keyword.

If you do not specify SIGNED, ABSVAL, ZONE, or DIGIT for a key field, the value of the field is treated as a string of unsigned binary data.

The following example shows how to specify the DIGIT keyword:



In this example, if ORDTYP is a 3-byte field, the values of the field for three different records could be as follows:

Values	Hexadecimal	Digits Used for Key
C4J	C3F4D1	341
LMA	D3D4C1	341
3D1	F3C4F1	341

EDTCDE(edit-code [* | floating-currency-symbol])

Use this field level keyword to specify an edit code for editing of output data using this current field description. The edit code specification is not used by data base data management, but is retained in the data base file description so that it can be duplicated into a device file description through the reference function at device file creation. An edit code is a required parameter value for this keyword.

EDTCDE is valid for numeric fields only. You cannot specify both the EDTCDE and the EDTWRD keywords for the same field. You can use the EDTCDE keyword to perform edit functions, such as the following:

- Suppress leading zeros
- Omit a sign from the low-order position of the field
- Punctuate the field without having to specify an edit word (see the EDTWRD keyword)
- Perform asterisk fill
- Specify floating currency symbol
- Perform user-defined edit functions

CPF Edit Codes: The edit codes that you can specify as a parameter value for this keyword are:

1 through 4
A through D
J through M
X through Z

See the EDTCDE keyword descriptions in Chapter 4, *Display Files*, and Chapter 5, *Printer Files*, for summaries of the functions provided by all CPF edit codes except X. Edit code X is used only to override the default edit codes used in query (default edit code is J) and the data file utility (default edit code is L).

User-Defined Edit Codes: You can also specify any of five user-defined edit codes (5 through 9). Before creating a device file from DDS source containing user-defined edit codes, the user-defined edit codes must exist. When you create the device file, the editing information is extracted from the user-defined edit codes as referenced in the DDS. Changing user-defined edit codes does not affect existing device files unless the device files are re-created. See the *CPF Programmer's Guide* and the *CL Reference Manual* for descriptions of how to create user-defined edit codes if those shipped with the system cannot be used.

Asterisk Fill and Floating Currency Symbol: You can specify asterisk fill or floating currency symbol with edit codes 1 through 4, A through D, and J through M.

When you specify asterisk fill, an asterisk (*) is written for each zero suppressed. A complete field of asterisks is printed for a zero balance field.

When you specify a floating currency symbol, the symbol appears to the left of the first significant digit. The symbol does not print on a zero balance when an edit code is used that suppresses the zero balance. The symbol that you specify must match the system value for the floating currency symbol (QCURSYM) when the file is created; it does not have to match when the file is used. See the *CPF Programmer's Guide* for a description of how to change this symbol.

The following example shows how to specify the EDTCDE keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B/A/S/X/Y/N/I/M)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A										PRICE		5		2				EDTCDE(J)
	A																		
00020	A										SALES		7		2				EDTCDE(K $)
	A																		
00030	A										SALARY		8		2				EDTCDE(L *)
	A																		
	A																		

See the EDTCDE keyword descriptions in Chapter 4, *Display Files*, and Chapter 5, *Printer Files*, for examples of valid edit codes, source data, and edited output.

EDTWRD('edit-word')

If you cannot accomplish the desired editing by using the EDTCDE keyword, you can specify an edit word instead. The edit word specification is not used by data base data management, but is retained in the data base file description so that it can be duplicated into device file descriptions through the reference function at device file creation. The edit word specifies the form in which the field values are to be displayed and clarifies the data by inserting characters directly (such as decimal points, commas, floating and fixed currency symbols, and credit balance indicators). The edit word can also be used to suppress leading zeros and provide asterisk-fill protection.

The following rules apply when you specify an edit word:

- The edit word must be enclosed in apostrophes.
- The EDTWRD keyword is valid for numeric fields only.
- The number of replaceable characters in the edit word must equal the length of the field.
- When you are using the floating currency symbol, the sum of the number of blanks and the stop-zero-suppression character (digit positions) contained in the edit word must be equal to the number of positions in the field to be edited. The currency symbol is not counted as a digit position.

  For example, if you specify a field length of 7 with 2 decimal positions, the edit word must be specified as:

  EDTWRD('b b b b $0. b b')

  where the b represents a blank.

- Any printable character is valid, but the following characters in certain positions have special meanings:

  – *Blank*: A blank is replaced with the character from the corresponding position of the data field. A blank position is referred to as a digit position.

  – *Ampersand*: An ampersand causes a blank in the edited field. The ampersand is not printed.

  – *Zero*: A zero stops zero suppression; place a zero in the rightmost position where zero suppression is to stop. The zero is replaced with the character from the corresponding position of the data field, unless that character is zero. Any zeros in the data that appear to the right of the stop-zero-suppression character are printed. At least one leading zero is suppressed. Each zero that is suppressed is replaced by a blank. The stop-zero-suppression character is considered a digit replace position.

- *Asterisk*: An asterisk stops zero suppression and replaces zeros with asterisks (asterisk protection). Place the asterisk in the rightmost position where zero suppression is to stop. Each zero that is suppressed is replaced by an asterisk. An asterisk preceding a zero is interpreted as representing asterisk protection; in this case, the zero prints as a constant. An asterisk is considered a digit replace position.

- *Currency Symbol*: A currency symbol coded immediately to the left of the zero suppression code causes the insertion of a currency symbol in the position to the left of the first significant digit. It is called the *floating currency symbol* when used in this manner.

  A currency symbol coded in the leftmost position of the edit word is fixed and prints in the same location each time. When used in this manner, it is called the *fixed currency symbol.*

  The currency symbol is not considered a digit replace position. This symbol must correspond to the QCURSYM system value.

- *Decimals* and *Commas*: Decimals and commas are printed in the same relative positions in which they are coded in the edit word unless they are to the left of the first significant digit. In that case, those positions are filled with blanks or replaced by asterisks.

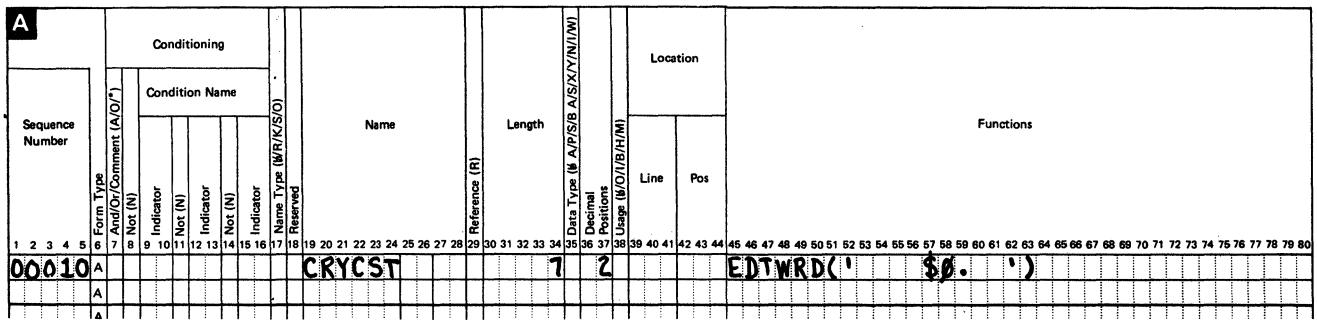
- *All Other Characters*: All other characters are printed if they are to the right of the significant digits in the data field. If they are to the left of the high-order significant digits in the data, those positions are filled with blanks or replaced by asterisks if asterisk protection is being used.

- If you want to show a negative sign with a negative number, include a sign in the edit word. Use either the minus sign (-) or the letters CR (credit). These print only if the number is negative.

- You cannot specify both the EDTWRD and EDTCDE keywords for the same field.

See the EDTWRD keyword description in Chapter 4, *Display Files*, for examples of edit words, source data, and edited output.

The following example shows how to specify the EDTWRD keyword:



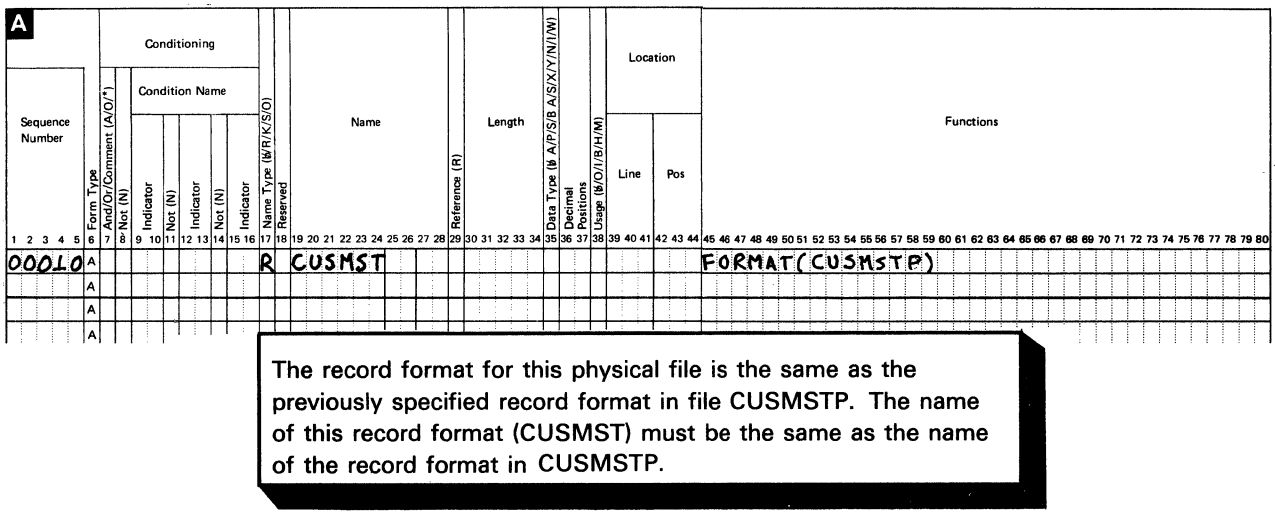
FORMAT(data-base-file-name[.library-name])

Use this record level keyword to specify that this record format is to share the specifications for a previously defined record format. The name of the record format you are defining must be the name of the previously defined record format. The data base file name is required; it is the name of the physical or logical file from which the previously defined record format is taken. The library name is optional; if you do not specify it, the library list in effect at file creation time is used.

If you specify the FORMAT keyword, you cannot specify field specifications for this record format; however, you must specify key specifications if you want them to be in effect for this file. (They can be the same as or different from the previously defined record format.)

If the data base file from which you are using the record format is deleted, the record format remains in existence as long as some file is using the record format. For example, RECORD in FILE2 uses the FORMAT keyword to share the specifications of RECORD in FILE1. Both files have been created. If FILE1 is deleted and then re-created with different DDS, RECORD2 still exists and can itself be referenced for the original record format by other files using the FORMAT keyword.

The following example shows how to specify the FORMAT keyword:



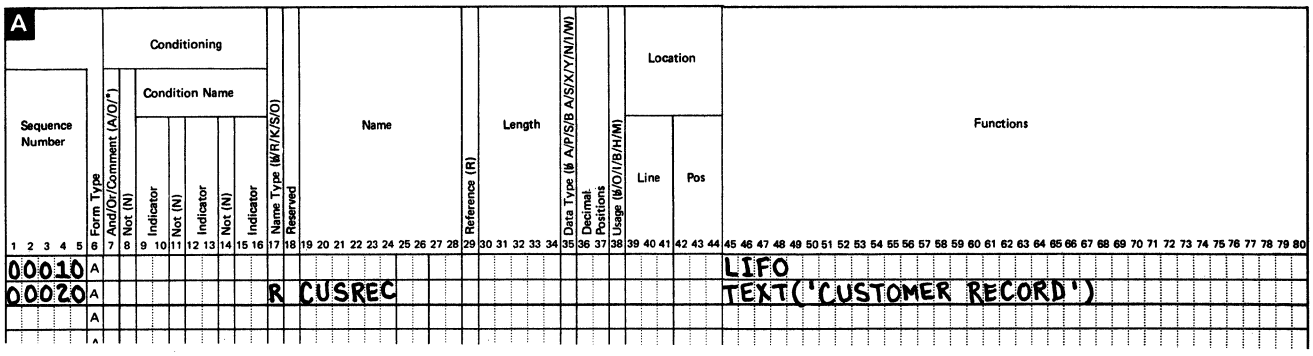
LIFO

Use this file level keyword to specify that if records with duplicate key values are retrieved from the same physical file member, they are to be retrieved in a last-in-first-out (LIFO) order.

If neither LIFO nor UNIQUE is specified, records that have duplicate key values are retrieved in a first-in-first-out (FIFO) order.

The LIFO keyword is not valid when FILETYPE(*SRC) is specified on the Create Physical File (CRTPF) command. See the *CPF Programmer's Guide* for more information about the FILETYPE(*SRC) parameter value.

The following example shows how to specify the LIFO keyword:



NOALTSEQ

Use this key field level keyword to specify that the ALTSEQ keyword specified at the file level is not to apply to this key field. If you specify ABSVAL or SIGNED for a key field, NOALTSEQ is in effect whether or not the NOALTSEQ keyword is specified for that key field.

The following example shows how to code the NOALTSEQ keyword. In this example, records with the record format DISTR are sequenced by the composite keys CODE and NAME, in which CODE is sequenced by the alternate collating sequence TABLE1 in TABLELIB, and NAME is sequenced by the EBCDIC collating sequence. NOALTSEQ prevents the sequence of the NAME field from being altered.

A

Sequence Number	Form Type	Name Type	Name	Length	Functions
00004	A				ALTSEQ(TABLE1.TABLELIB)
00005	A	R	DISTR		
00006	A		{		
00007	A		}		
00008	A		CODE	1	
00009	A		NAME	2Ø	
00010	A		{		
00011	A		}		
00012	A	K	CODE		
00013	A	K	NAME		NOALTSEQ
	A				
	A				

RANGE(low high)

This field level keyword is a validity checking keyword for display files. RANGE is not used by data base data management. It is specified in the data base file description so that it can be duplicated into display file descriptions when this field is referenced at display file creation. The rules for specifying this keyword are the same for a physical file as for a display file. See Chapter 4, *Display Files*, for information on how to specify this keyword.

REF(data-base-file-name[.library-name][record-format-name])

Use this file level keyword to specify the name of a file from which field descriptions are to be retrieved. Use it when you want to use field attributes from several fields in a previously described record format for the file being defined. You can specify the file name once in the REF keyword instead of specifying it on several REFFLD keywords with each of the field descriptions that reference the file. To reference more than one file, use the REFFLD keyword. (The REF keyword can be specified only once.)

If there is more than one record format in the referenced file, you can specify a record format name as a parameter value for this keyword to tell CPF which one to use, unless the formats should be searched sequentially.

The file name is a required parameter value for this keyword; the library name and the record format name are optional.

If you do not specify the library name, the current library list is used. If the record format name is not specified, each record format is searched in order (as they are specified); the first occurrence of the field name is used. For information on how the choice of REF and REFFLD keywords controls these searches, see Appendix A, *How to Specify REF and REFFLD.*

The following examples show how to specify the REF keyword:

Sequence Number	Form Type	Name Type (b/R/K/S/O)	Name	Reference (R)	Functions
00010	A				REF(FILE1)
00020	A	R	RECORD		
00030	A		FLD1	R	
	A				
	A				
	A				
	A				
	A				
	A				
	A				
	A				

In this example, FLD1 has the same attributes as the first (or only) FLD1 in FILE1.

Sequence Number	Form Type	Name Type (b/R/K/S/O)	Name	Reference (R)	Functions
00010	A				REF(FILE1.LIB1 RECORD2)
00020	A	R	RECORD		
00030	A		FLD1	R	
	A				
	A				
	A				
	A				
	A				
	A				
	A				
	A				

In this example, FLD1 has the same attributes as FLD1 in RECORD2 in FILE1 in LIB1.

REFFLD(referenced-field-name[.record-format-name] [{*SRC | data-base-file-name[.library-name]}])

Use this field level keyword when referencing a field under one of these three conditions:

- The name of the referenced field is different from the name in positions 19 through 28.
- The name of the referenced field is the same as the name in positions 19 through 28, but the record format, file, or library of the referenced field is different from that specified with the REF keyword.
- The referenced field occurs in the same DDS source file as the referencing field.

The referenced field name is required even if it is the same as the referencing field. Use the record format name when the referenced file contains more than one record format. Use *SRC to search the DDS source file in which the referencing field occurs (this is the default value when the data base file name and the library name are not specified). Specify the data base file name (qualified by its library name, if necessary) when you want to search a particular data base file.

If, in the same DDS source file, you specify REF at the file level and REFFLD at the field level, the REFFLD specification is used. The particular search sequence depends on both the REF and REFFLD keywords. For more information, see Appendix A, *How to Specify REF and REFFLD*.

An R must be specified in position 29.

The following example shows how to code the REFFLD keyword. Because the REF keyword is not specified, the default for line 00030 and 00040 is to search the DDS source file in which they are specified. In line 00080, the parameter *SRC explicitly specifies the source file. See the example in Appendix A, *How to Specify REF and REFFLD*, for explanations of the various specifications.

A

Sequence Number (1-5)	Form Type (6)	Name Type (17)	Name (19-28)	Reference (29)	Length (30-34)	Location Pos (42-44)	Functions (45-80)
00010	A	R	RECORD1				
00020	A		ITEM		5	1	
00030	A		ITEM1	R		2	REFFLD(ITEM)
00040	A		ITEM2	R		12	REFFLD(ITEM.FMAT1)
00050	A		ITEM3	R		22	REFFLD(ITEM FILEX)
00060	A		ITEM4	R		32	REFFLD(ITEM FILEX.LIBY)
00070	A		ITEM5	R		42	REFFLD(ITEM.FMAT1 FILEX.LIBY)
00080	A		ITEM6	R		52	REFFLD(ITEM *SRC)
	A						
	A						
	A						
	A						
	A						
	A						

SIGNED

This key field level keyword specifies that when sequencing the values associated with this numeric key field, CPF is to consider the signs of the values (negative versus positive values). SIGNED is not valid for a field of character data type.

The following example shows six records with zoned decimal key fields:

Record	Numeric Key Field (Zoned Decimal)	Hexadecimal Representation
1	98	F9F8
2	00	F0F0
3	98-	F9D8
4	97	F9F7
5	20	F2F0
6	99	F9F9

If no sequencing keywords are specified, the records are sequenced in this order:

Record	Numeric Key Field (Zoned Decimal)	Hexadecimal Representation
2	00	F0F0
5	20	F2F0
3	98-	F9D8
4	97	F9F7
1	98	F9F8
6	99	F9F9

If SIGNED is specified, the records are sequenced in this order:

Record	Numeric Key Field (Zoned Decimal)	Hexadecimal Representation
3	98-	F9D8
2	00	F0F0
5	20	F2F0
4	97	F9F7
1	98	F9F8
6	99	F9F9

If both SIGNED and DESCEND are specified, the records are sequenced in this order:

Record	Numeric Key Field (Zoned Decimal)	Hexadecimal Representation
6	99	F9F9
1	98	F9F8
4	97	F9F7
5	20	F2F0
2	00	F0F0
3	98-	F9D8

Note: SIGNED is the same for logical and physical files.

SIGNED (a key field level keyword) and ALTSEQ (a file level keyword) are mutually exclusive. If you specify SIGNED for a key field, NOALTSEQ is in effect for that key field even if ALTSEQ is specified at the file level (whether NOALTSEQ is specified or not).

If you do not specify SIGNED, ABSVAL, ZONE, or DIGIT for a key field, the value of the key is treated as a string of unsigned binary data.

The following example shows how to specify the SIGNED keyword:

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00010	A								R		RECORDA								
00020	A										FLDA		7	S	2				
00030	A										FLDB								
00040	A								K		FLDA								SIGNED
	A																		
	A																		
	A																		

TEXT('description')

Use this record or field level keyword to supply a text description with this record format. The text description must be enclosed in apostrophes, with a maximum of 50 characters. If more than 50 characters are specified, the keyword is accepted; however, the excess characters are dropped and a warning message is issued.

You can specify TEXT at both the record format level and the field level. However, both TEXT and FORMAT cannot be specified at the record format level, because the format already contains text.

The following examples show how to specify the TEXT keyword:

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00010	A										ORDNBR		5		0				COLHDG('ORDER' 'NUMBER')
00020	A																		TEXT('ORDER NUMBER')
	A																		
	A																		

If TEXT is not specified and COLHDG is specified, 50 positions of column heading information are used as a default for text. If COLHDG is not specified, there is no text, unless it is retrieved from a referenced field.

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9–10)	Not (N) (11)	Indicator (12–13)	Not (N) (14)	Indicator (15–16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19–28)	Reference (R) (29)	Length (30–34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36–37)	Usage (b/O/I/B/H/M) (38)	Location Line (39–41)	Location Pos (42–44)	Functions (45–80)
00070	A										YTDAMT		6		0				COLHDG('Year to Date' 'Amount')
	A																		
	A																		
	A																		

If the TEXT keyword is not specified, the information specified on the COLHDG keyword is used for the text as if coded as follows:

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9–10)	Not (N) (11)	Indicator (12–13)	Not (N) (14)	Indicator (15–16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19–28)	Reference (R) (29)	Length (30–34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36–37)	Usage (b/O/I/B/H/M) (38)	Location Line (39–41)	Location Pos (42–44)	Functions (45–80)
	A																		
00070	A										YTDAMT		6		0				TEXT('Year to Date Amount')
	A																		
	A																		
	A																		

UNIQUE

Use this keyword at the file level to specify that records with duplicate key values are not allowed within a member of this physical file. Any inserts or additions of new records, or updates to existing records, that would result in a duplicate key are rejected. The application program issuing the write or the update operation receives a message. When a work station user is using DFU, a message is displayed at the work station. A copy file command copying records that would have duplicate keys in this file cannot be completed.

When a logical file based on this physical file has the UNIQUE keyword, the physical file member or members cannot have records with duplicate keys.

When the UNIQUE keyword is specified for a physical file, the MAINT(*IMMED) parameter value must be specified on the Create Physical File (CRTPF) command that creates the file. This means that the access path is maintained immediately as changes are made. See the *CPF Programmer's Guide* for a description of the MAINT(*IMMED) parameter value.

If the UNIQUE keyword is not specified, records with duplicate key values are sequenced in first-in-first-out (FIFO) order, unless the LIFO keyword is specified, in which case they are sequenced in last-in-first-out order. For an explanation of how records with duplicate key values are sequenced when those records are not in the same file member, see the section on data base records in the *CPF Programmer's Guide*.

The following example shows how to specify the UNIQUE keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (b/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (b A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (b/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A	*																	
00020	A	*		SAMPLE PHYSICAL FILE (CUSMSTP2)															
00030	A	*																	
00040	A																		UNIQUE
00050	A								R		CUSMST2								FORMAT(CUSMSTP)
00060	A								K		NAME								
00070	A								K		ADDR								
	A																		
	A																		
	A																		

VALUES(value-1 [value-2 . . .[value-20]])

This field level keyword is a validity checking keyword for display files. It does not request any action by data base data management, but can be specified in the physical file so that it can be duplicated into display file descriptions when this field is referenced at display file creation. The rules for specifying this keyword are the same for a data base record format as for a display file record format. See Chapter 4, *Display Files*, for information on how to specify this keyword.

ZONE

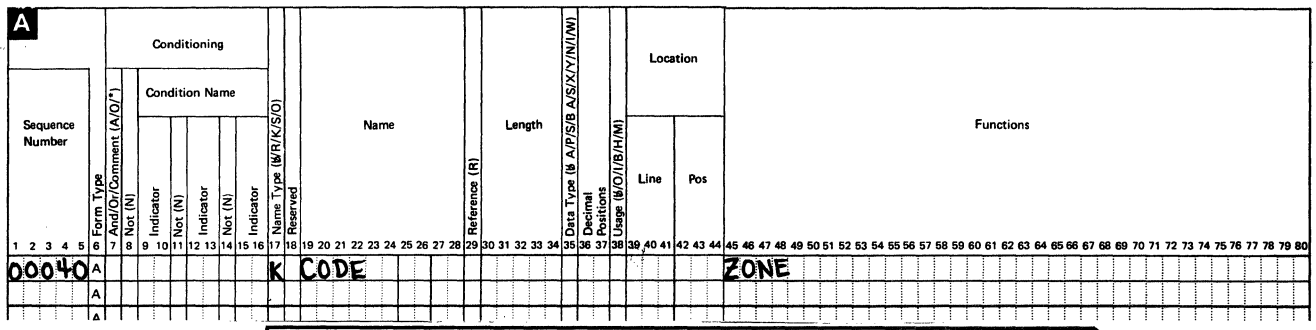
Use this key field level keyword to specify that only the zone portion (leftmost four bits) of each byte of the key field is to be used when constructing a value associated with this key field. The digit portion is filled with zeros.

This keyword is applied against the entire key field (not just a position within the field), and it is valid only for character or zoned decimal type fields.

You cannot use ZONE with the ABSVAL, SIGNED, or DIGIT keywords.

If you do not specify SIGNED, ABSVAL, ZONE, or DIGIT for a key field, the value of the field is treated as a string of unsigned binary data.

The following example shows how to specify the ZONE keyword:



In this example, if CODE is a 1-byte field, the values of the field for three different records could be as follows:

Values	Hexadecimal	Zone Used for Key
A	C0	C
B	C1	C
E	C4	C

Chapter 3. Logical Files

This chapter is divided into two sections. The first, *Positional Entries*, provides rules and examples for filling in positions 1 through 44 of the Data Description Specifications form. To find information in this section, first determine what position on the form to use, then look up the section describing that position.

The second section of this chapter, *Keyword Entries*, provides rules and examples for specifying DDS keywords. The keywords are described in alphabetical order.

For guidance in choosing positional entries and keywords for logical files, see the *CPF Programmer's Guide*.

DEFINING A LOGICAL FILE

To define a logical file, specify the entries in the following order:

1. File level entries (optional)
2. Record level entries
3. Field level entries (optional)
4. Key field level entries (optional)
5. Select/omit level entries (optional)

} Repeat these entries for each record format in the file.

Note: The file name is provided through the Create Logical File (CRTLF) command, not through DDS.

Complete logical file examples can be found in Appendix F, *Examples*.

An explanation of *file level*, *record level*, *field level*, *key field level*, and *select/omit field level* can be found in Chapter 1, *Introduction*.

The syntax rules for specifying DDS keywords can be found in Chapter 1, *Introduction*.

For the maximum number of record formats in a logical file, see the PFILE keyword description in the *Keyword Entries* section in this chapter. The maximum number of fields in any one record format is 8000; the maximum number of bytes in any one record format is 32 766.

SPECIFYING MORE THAN ONE RECORD FORMAT IN A LOGICAL FILE

When you specify more than one record format in a logical file, each record format requires the PFILE keyword. In a logical file, there are three ways to specify the fields to be used in any record format, as follows:

- Specify only the record format name and the PFILE keyword.
- Specify only the record format name, the PFILE keyword, and the FORMAT keyword.
- Specify the record format name, the PFILE keyword, and individual fields as needed.

Each of the above ways of specifying the fields in a record format can also have the following access path specifications:

- Specify no key fields and no select/omit fields (arrival sequence access path; only one record format with one based-on physical file can be specified for the logical file).
- Specify one or more key fields (keyed sequence access path; all record formats in the logical file must have at least one key field specified).
- Specify the ACCPTH keyword (keyed sequence access path; the file you are defining shares an access path defined by another physical or logical file).

In addition, in any logical file with a keyed sequence access path, select/omit fields can be specified for any number of the record formats in the file.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050*
Printed in U.S.A.

File		Keying Instruction	Graphic								Description	Page	of
Programmer	Date		Key										

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A	*		LOGICAL					F		ILE EXAMPLE								
00020	A	*		INVENTORY							FORMAT								
00030	A								R		INVFMT								PFILE(INVENTORY)
00040	A								K		ITEM								
00050	A	*																	
00060	A	*		ORDER		FORMAT													
00070	A								R		ORDFMT								PFILE(ORDER)
00080	A																		TEXT('ORDER ANALYSIS')
00090	A										ITEM								
00100	A										ORDER		10						
00110	A										SUPPLY		+2						
00120	A										SHPDAT								CONCAT(SHPMO SHPDA SHPYR)
00130	A										QTY		5	P					RENAME(QTYDUE)
00140	A								K		ITEM								
00150	A								K		SHPYR								
00160	A								K		SHPMO								
00170	A								K		SHPDA								
00180	A								O		QTYDUE								CMP(LT 1)
00190	A	*																	
00200	A	*		ACCOUNTING		FORMAT													
00210	A								R		ACTFMT								PFILE(ACCOUNTS)
00220	A																		FORMAT(ACCOUNTL)
00230	A								K		ITEM								
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

*Number of sheets per pad may vary slightly.

Figure 3-1. Logical File Coding Example

Positional Entries (Positions 1-44)

This section describes how to specify the first 44 positions of the Data Description Specifications form for logical files. To code the rest of the form, see the section *Keyword Entries (Functions Field, Positions 45-80)* in this chapter.

Figure 3-1 shows some positional entries for logical files.

Positions 1-5 (Sequence Number)

Use these positions to specify a sequence number for each line on the form. The sequence number is optional and is used for documentation purposes only.

Position 6 (Form Type)

The A in this position designates this as a Data Description Specifications form. The form type is optional and is for documentation purposes only.

Position 7 (Comment)

An asterisk (*) in position 7 identifies the line as a comment line. Comments can appear anywhere in DDS and are retained only in the source file and printed on the source listing. (Comments are not printed on the expanded source listing.) Use positions 8 through 80 for comment text. A blank line (no characters specified in positions 7 through 80) is treated as a comment line.

Positions 8-16 (Conditioning)

These positions do not apply to logical files and must be blank unless an asterisk appears in position 7 (indicates a comment line).

Position 17 (Name Type)

The value in this position identifies the type of name specified in positions 19 through 28 (name). The valid entries are:

Entry	Meaning
R	Record format name
Blank	Field name or select/omit AND condition
K	Key field name
S	Select field name
O	Omit field name

Position 18 (Reserved)

This position does not apply to any file and must be blank unless an asterisk appears in position 7 (indicates a comment line).

Positions 19-28 (Name)

Use these positions to specify the names of the following:

- The record format or formats for this logical file
- For each record format, the field or fields that make up the record format (unless you specify the FORMAT keyword at the record level)
- For each record format, the field or fields to be used as key fields
- For each record format, the field or fields to be used for select/omit specifications

When specifying names in DDS, the following rules apply:

- Names must be 10 characters or less.
- Names must start in position 19.
- The name must begin with an alphabetic character (A through Z, @, $, and #) or asterisk (*) (for *NONE). All subsequent characters can be alphameric (A through Z, 0 through 9, @, $, #, and underscore (_)). There can be no embedded blanks.
- The type of name must be specified in position 17, unless you are specifying a field name or select/omit AND condition.

Figure 3-1 shows how to code the name field.

High-level languages can impose specific length and value restrictions on the name field. For example, RPG III accepts only file names of 8 characters or less, record format names of 8 characters or less, and field names of 6 characters or less, and they cannot contain an underscore. It is your responsibility to ensure that the name syntax used here is acceptable to all language processors that process the file.

When you specify R in position 17 (name type), the name specified in positions 19 through 28 is a record format name. More than one record format name can be specified for a logical file, but each must be unique within that file. (See the appropriate high-level language manual for exceptions.) Specify the record format name in one of three ways:

- As the record format name in the first physical file specified on the PFILE keyword. This is required if you do not specify the FORMAT keyword and if you do not define individual fields in this record format.

- As the name of a *new* record format with field names specified in this logical file. The name of the record format can be the same as the file name specified in the create file command; however, a warning message is issued if the names are not unique, because some language processors do not allow record format and file names to be the same. RPG III is such a language.

- As the name of a record format *previously described* in a physical or logical file. Field names and attributes are not specified and the FORMAT keyword must be specified. The FORMAT keyword is explained in the *Keyword Entries* section in this chapter.

Use the PFILE keyword in conjunction with the record format name to specify the physical file(s) with which the record format is to be associated. A record format can be used with more than one physical file. If no fields are defined and the FORMAT keyword is not specified, the format of the first file specified in the PFILE keyword is used as the format for all the physical files. (This format is used for field attribute references and attribute and name checking.)

Field Name

When position 17 (name type) is left blank, the name specified in positions 19 through 28 is a field name. Each field in a logical file must be named. The order in which you specify the field names is the order in which the fields appear to programs that use the logical file. Each field name must be unique within the record format and must correspond to a field in the physical file record format.

The name you give to a field in a logical file record format will usually be the same as the name of the corresponding field in the physical file record format. If different, the two names must be equated by using the RENAME keyword. A field in a logical file record format can also represent the concatenation of two or more fields from the physical file. (See the CONCAT keyword.)

Note: The sequence in which the field names are specified in the logical file is important. If the same physical field is specified more than once in a record format in the logical file (that is, by using either RENAME or CONCAT), the sequence in which the fields are specified in the logical file is the sequence in which the data is moved to the physical file. Thus, the value of the field the last time the field is specified in the logical file is the value in the physical record. If a renamed (RENAME keyword) or concatenated (CONCAT keyword) field is used as a key field, the value of the key for the purposes of ordering the record is the value assigned to the first occurrence of the field in the record format.

When you specify K in position 17 (name type), the name specified in positions 19 through 28 is a *key field name.* Use key fields (and optionally, select/omit fields) to define a keyed sequence access path for record formats in the logical file member. (The logical file member includes the physical file members specified on the DTAMBRS parameter on the Create Logical File (CRTLF) or Add Logical File Member (ADDLFM) commands.)

If you specify more than one record format for a logical file or more than one physical file for the PFILE keyword, you must specify at least one key field for all record formats of that logical file. A key can have more than one key field; such a key is a *composite key.*

The following example shows a logical file with two record formats, one of which uses a composite key:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		RECORD1								PFILE(PF1)
00020	A										FIELD1								
00030	A										FIELD2								
00040	A										FIELD3								
00050	A								K		FIELD1								
00060	A	*																	
00070	A								R		RECORD2								PFILE(PF2)
00080	A										FIELD4								
00090	A										FIELD5								
00100	A								K		FIELD4								
00110	A								K		FIELD5								
	A																		
	A																		
	A																		

In this example, RECORD1 has a single key field, FIELD1. RECORD2 has a composite key that includes FIELD4 and FIELD5.

In a composite key, specify the key field names in the order of importance (major to minor), and specify each key field name on a separate line.

If you specify no key field, the logical file you are defining has an arrival sequence access path. Also, without at least one key field specified, you cannot specify select/omit fields.

The number of fields that make up a key is restricted in that the total key length cannot exceed 120 bytes. The total key length includes the length of each key field. When you specify more than one record format in a logical file, add 1 byte for each key field (CPF uses the extra byte when records from different physical files have duplicate key values).

Fields that you specify as key fields must exist in the logical file record format as well as in the physical file record format, and the key field name must be the name specified for the field in the physical file record format. Data type, length, and decimal positions of the field in the physical file record format cannot be overridden in the logical file record format; they must be the same as in the physical file record format. The field used as a key field must exist at the field level in the record format as well as at the key field level. If you do not specify field names at the field level, the field must exist in the record format specified with the FORMAT keyword or in the record format of the first physical file specified with the PFILE keyword.

You can specify one or more *access path keywords* to affect the way CPF builds and uses key values. The access path keywords are the following:

File Level

ACCPTH
ALTSEQ
LIFO
UNIQUE

Key Field Level

DESCEND
DIGIT
NOALTSEQ
SIGNED
ZONE

Different key fields within a composite key can have different access path keywords.

Logical Files with More Than One Record Format: When you specify more than one record format in a logical file, you must specify at least one key field for every record format in the logical file. It is not necessary to specify the same number of key fields in each key. Also, key fields specified in one record format must have the same field attributes and access path keywords as the corresponding key fields in other record formats in the same logical file.

A key is required for every record format so that the logical file member can have a single access path sequencing records of each record format. When records are returned from the various based-on physical file members, they are merged according to the values of the key fields in the access path for the logical file member.

When records of a logical file member are sequenced, CPF builds a *key value* for each record by concatenating the values in its key fields. The key value is then used to build the access path for use by your program. See the section on data base records in the *CPF Programmer's Guide* for information about the I/O operations permitted by CPF.

Each key field in a composite key has a *key position.* The first key field specified is in position 1, the second key field specified is in position 2, and so on. During I/O operations to a logical file, CPF compares the key values of the records written to or read from the data base. When you create a logical file that has more than one record format (with or without different key fields specified), CPF performs *key position attribute checking.* For key position attribute checking to succeed, key fields of different record formats that are in the same key positions must have the same data type, length, decimal positions, and access path keywords specified at the key field level. This ensures a meaningful record sequence from the comparisons made during an I/O operation. For example, in the following logical file, FIELD1 and FLD1 must have the same attributes, and FIELD2 and FLD2 must have the same attributes. FIELD1 and FLD1 are in key position 1, and FIELD2 and FLD2 are in key position 2.

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B/A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		RECORD1								PFILE(PF1)
00020	A										FIELD1		5		0				
00030	A										FIELD2		10						
00040	A										FIELD3		10						
00050	A								K		FIELD1								
00060	A								K		FIELD2								DESCEND
00070	A	*																	
00080	A								R		RECORD2								PFILE(PF2)
00090	A										FLD1		5		0				
00100	A										FLD2		10						
00110	A										FLD3		20						
00120	A								K		FLD1								
00130	A								K		FLD2								DESCEND
	A																		
	A																		
	A																		

For examples of key fields in a logical file with more than one record format, refer to Figure 3-1. In Figure 3-1, fields named ITEM are specified in each key. For record formats INVFMT and ACTFMT, ITEM is the only key field specified. For record format ORDFMT, a composite key is specified. This composite key includes ITEM, SHPYR, SHPMO, and SHPDA. Each of the fields used in a key must also exist at the field level; therefore, ITEM must exist in the record format for the physical file INVENTORY so that it can be copied into this logical file for INVFMT. Also, ITEM must exist in the record format for the logical file ACCOUNTL so that it can be copied into this logical file for ACTFMT. (ITEM must also exist in physical file ACCOUNTS.)

*How to Use *NONE:* Two conditions can occur in which key fields having the same key position should not be compared. The two conditions are:

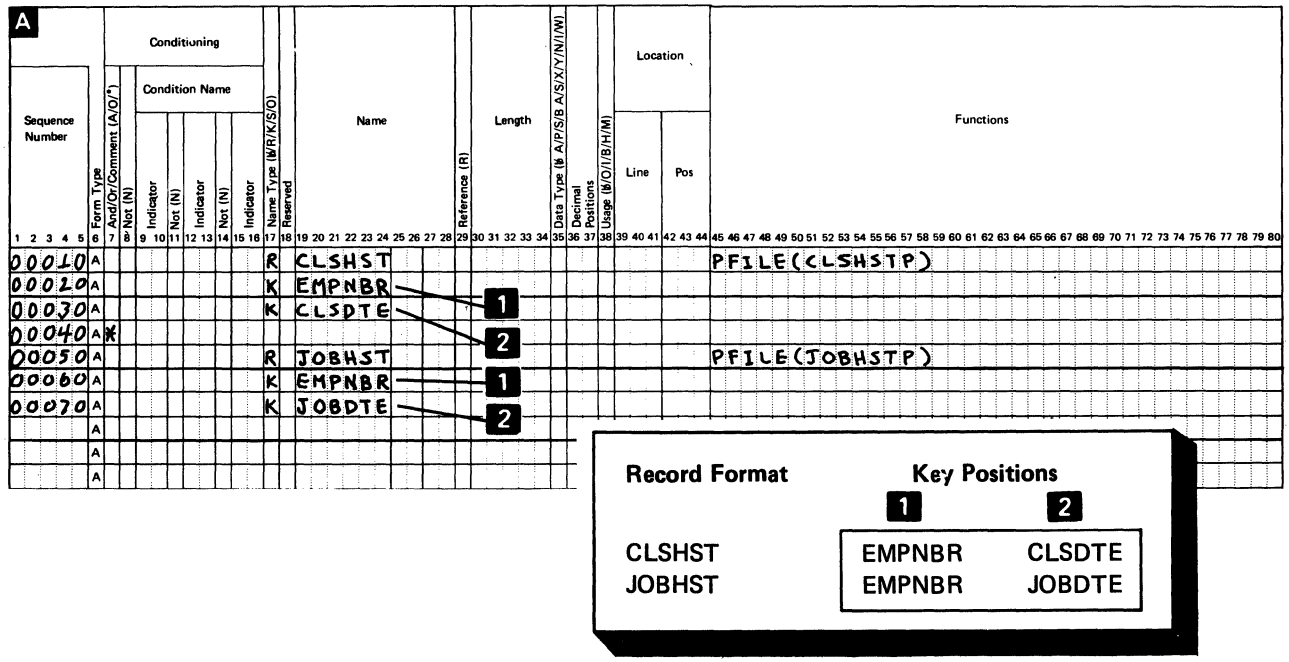
- The key fields do not have the same field attributes (data type, length, decimal positions, or access path keywords at the field level).
- The key fields have the same attributes, but you do not want them to be merged and sequenced together.

To avoid invalid or unwanted comparisons between key fields, specify *NONE in place of one of them, and move the displaced key field to the next key position. CPF compares the values of key positions before and after *NONE, but retrieves the affected records in the order in which the record formats are specified in the DDS for the logical file.

Note: You can specify *NONE two or more times on succeeding lines to displace a key field to a key position for which a comparison of key field attributes is relevant to your application.

Example 1: A logical file views records of two physical files through two different record formats, CLSHST (class history) and JOBHST (job history). In the logical file, the records from the two physical files can be merged together and sequenced by employee identification number (EMPNBR) by specifying EMPNBR in key position 1.

All records that have the same key value for EMPNBR pertain to the same employee. To merge and sequence all records for a given employee into a single history of classes and job assignments, specify CLSDTE (date of class) and JOBDTE (date of job assignment) in key position 2 for the two record formats, as follows:

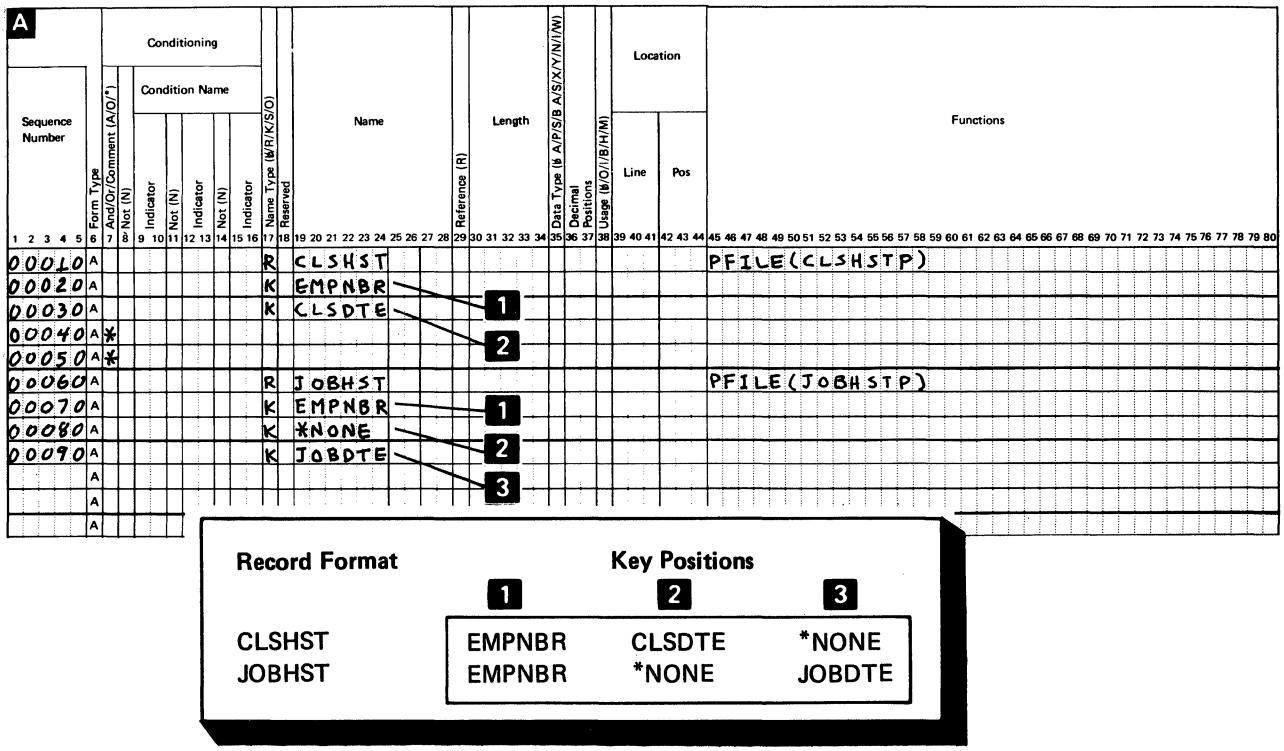


Suppose that the job assignment dates and class dates are the dates (month/year) that the class or assignment started. Records for three students are retrieved in the following order:

EMPNBR	CLSDTE	JOBDTE	Description
1005	3/79		Completed class
1005	4/79		Withdrew to begin new job
1005		4/79	Completed job
1005	6/79		Completed class
1006		1/79	Completed job
1006		2/79	Completed job
1006	3/79		Completed class
1006	5/79		Transferred to new location
1007		1/79	Completed job
1007		4/79	Completed job
1007		7/79	Completed job
1007	8/79		Withdrew because of illness

Note: Such a report provides a continuous history for each student.

Example 2: Another logical file views the same two physical files as in example 1, but the second record format in the logical file has *NONE specified in key position 2, as in the following DDS:



As in example 1, all records from the two physical files are first merged and sequenced together on employee number (EMPNBR); however, the records for each student are merged and sequenced first on class date (CLSDTE) and then on job assignment date (JOBDTE). The set of records used for example 1 are retrieved as follows:

EMPNBR	CLSDTE	JOBDTE	Description
1005	3/79		Completed class
1005	4/79		Withdrew to begin new job
1005	6/79		Completed class
1005		4/79	Completed job
1006	3/79		Completed job
1006	5/79		Transferred to new location
1006		1/79	Completed job
1006		2/79	Completed job
1007	8/79		Withdrew because of illness
1007		1/79	Completed job
1007		4/79	Completed job
1007		7/79	Completed job

When several adjacent record formats have *NONE in the same key position, they form a set, relative to record formats specified before and after them, that functions in sequencing as an individual record format. Key fields specified after *NONE serve to merge and sequence records of the formats within the set. The following example shows how several record formats function as a set.

Example 3: Consider a logical employee file over five physical files. The five record formats are defined as follows:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A								R		EMPMST								PFILE(EMPMSTP)
00020	A								K		EMPNBR		1						
00030	A	*																	
00040	A								R		CLSREG								PFILE(CLSREGP)
00050	A								K		EMPNBR		1						
00060	A								K		CLSDTE		2						
00070	A	*																	
00080	A								R		CLSHST								PFILE(CLSHSTP)
00090	A								K		EMPNBR		1						
00100	A								K		CLSDTE		2						
00110	A	*																	
00120	A								R		JOBHST		1						PFILE(JOBHSTP)
00130	A								K		EMPNBR		2						
00140	A								K		*NONE		2						
00150	A								K		JOBDTE		3						
00160	A	*																	
00170	A								R		ACTHST		1						PFILE(ACTHSTP)
00180	A								K		EMPNBR		2						
00190	A								K		*NONE		2						
00200	A								K		ACTDTE		3						
	A																		
	A																		
	A																		

Record Format	Key Positions 1	2	3
EMPMST	EMPNBR	*NONE	*NONE
CLSREG	EMPNBR	CLSDTE	*NONE
CLSHST	EMPNBR	CLSDTE	*NONE
JOBHST	EMPNBR	*NONE	JOBDTE
ACTHST	EMPNBR	*NONE	ACTDTE

The records are merged and sequenced as follows:

1. All records are merged and sequenced by employee number.

2. For a given employee, records are sequenced by:
   a. The master record (of the EMPMST format)
   b. Records of the CLSREG and CLSHST formats, merged and sequenced together on values of CLSDTE (key position 2)
   c. Records of the JOBHST and ACTHST formats, merged together and sequenced together on values of JOBDTE and ACTDTE (key position 3)

*NONE in the key definitions achieves this sequencing as follows:

- *NONE and a field name, CLSDTE, appear in the second key position of the adjacent formats, CLSHST and JOBHST. This effectively causes a split between the two formats after the preceding key position (position 1). Records of formats above the split are merged and sequenced with records of formats below the split only on values of EMPNBR.

- An implicit *NONE in the second key position of the format EMPMST forces a similar split.

- With *NONE in key position 2, the JOBHST and ACTHST formats form a set in which the values of JOBDTE and ACTDTE are compared in order to merge and sequence records of these two formats only.

Note that the record sequence defined by the previous key specifications is totally dependent on the order in which the formats are specified. For example, if JOBHST had been specified before CLSHST so that key position 2 reads

*NONE, CLSDTE, *NONE, CLSDTE, *NONE,

the values of CLSDTE within CLSREG would not have been sequenced with the values of CLSDTE within CLSHST, and JOBDTE would not have been sequenced with ACTDTE.

Example 4: Assume that an employee has repeated a class. To sequence two records with the same values for EMPNBR and CLSDTE, a third key field, DATE, is specified in record format CLSHST. However, DATE cannot be specified in the next available key position (position 3) because JOBDTE and ACTDTE appear in that position for other formats. If DATE is specified in this position, the attributes of DATE are compared with the attributes of CLSHST and JOBHST, and the key definitions are rejected.

To obtain the sequencing necessary, specify *NONE before DATE, displacing DATE to key position 4.

The DATE field can be shown in position 4 as follows:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B/A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		EMPMST								PFILE(EMPMSTP)
00020	A								K		EMPNBR								
00030	A	*																	
00040	A								R		CLSREG								PFILE(CLSREGP)
00050	A								K		EMPNBR								
00060	A								K		CLSDTE								
00070	A	*																	
00080	A								R		CLSHST								PFILE(CLSHSTP)
00090	A								K		EMPNBR								
00100	A								K		CLSDTE								
00110	A								K		*NONE								
00120	A								K		DATE	A							
00130	A	*																	
00140	A								R		JOBHST								PFILE(JOBHSTP)
00150	A								K		EMPNBR								
00160	A								K		*NONE								
00170	A								K		JOBDTE								
00180	A	*																	
00190	A								R		ACTHST								PFILE(ACTHSTP)
00200	A								K		EMPNBR								
00210	A								K		*NONE								
00220	A								K		ACTDTE								
00230	A																		

Record Format	Key Positions 1	2	3	4
EMPMST	EMPNBR	*NONE	*NONE	*NONE
CLSREG	EMPNBR	CLSDTE	*NONE	*NONE
CLSHST	EMPNBR	CLSDTE	*NONE	DATE (A)
JOBHST	EMPNBR	*NONE	JOBDTE	*NONE
ACTHST	EMPNBR	*NONE	ACTDTE	*NONE

Specifying DATE (A) in key position 4 enables records from physical file CLSHSTP with identical values for EMPNBR and CLSDTE to be merged and sequenced according to the value for DATE. See the *CPF Programmer's Guide* for detailed information regarding records with duplicate key values.

Select/Omit Field Name

Use select/omit fields to tell CPF how to select or omit records when your program retrieves them using this record format. The only records affected are those from the physical file(s) specified for the PFILE keyword for this record format. *If you do not specify any key field names, you cannot specify select/omit field names.*

When using select/omit fields, specify either S or O (or leave it blank for an AND condition) in position 17. In positions 19 through 28, specify a field name whose contents at execution time determine whether the record is to be selected or omitted based upon the select/omit keyword specified for this field. The select/omit keywords are COMP, RANGE, and VALUES. The last select/omit specification can be made with the ALL keyword, but no field name is permitted.

The field must appear in both the physical file record format and the logical file record format. The field can be renamed or be part of a concatenated field as long as it exists in the record format. The name you specify must be the one in the physical file. Select/omit statements must follow all field and key field level entries for the record format. Both select and omit can be specified for the same record format. The following information applies:

- If you specify both select and omit for a record format, the order in which they are specified is important. The select/omit statements are processed in the order they are specified; if a record satisfies a statement, the record is either selected or omitted as specified, and remaining select/omit statements are not examined.

- If you specify both select and omit statements, you can indicate whether records not meeting any of the criteria specified are to be selected or omitted. (See the ALL keyword.)

- If you do not specify the ALL keyword, the action taken for the records that do not meet the criteria is the converse of the type of the last statement specified. The records that do not meet selection criteria are omitted; records that do not meet omission criteria are selected.

There are limits to the number of select/omit statements you can specify in a single logical file. If you specify many select/omit statements and you cannot create the file, reduce the overhead for the file through the following changes in the specifications, in decreasing order of importance:

- Reduce the number of record formats in the file.

- Reduce the number of physical files specified on the PFILE keyword.

- Reduce the number of fields used (single occurrences) in the select/omit specifications.

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (b A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (b/O/I/B/H/M)	Location: Line	Location: Pos	Functions	
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	
00010	A								S		ST								COMP(EQ 'NY')	1
00020	A										REP								COMP(EQ 'JSMITH')	
00030	A										YEAR								COMP(LT 78)	
00040	A								O										ALL	
	A																			
00050	A								O		YEAR								COMP(GE 78)	2
00060	A								S		ST								COMP(EQ 'NY')	
00070	A										REP								COMP(EQ 'JSMITH')	
00080	A								O										ALL	
	A																			
00090	A								O		REP								COMP(NE 'JSMITH')	3
00100	A								O		ST								COMP(NE 'NY')	
00110	A								S		YEAR								COMP(LT 78)	
00120	A								O										ALL	
	A																			
	A																			

In this example, you want to select all the records before 1978 for a sales representative named JSMITH in the state of New York. There are three ways to code this example.

1 All records must be compared with the select fields ST, REP, and YEAR before they can be selected or omitted.

2 All records in and after 1978 are omitted in the first comparison. Then, only the records before 1978 are compared with ST and REP. Only two select fields must be satisfied. This way is more efficient than the way used in 1.

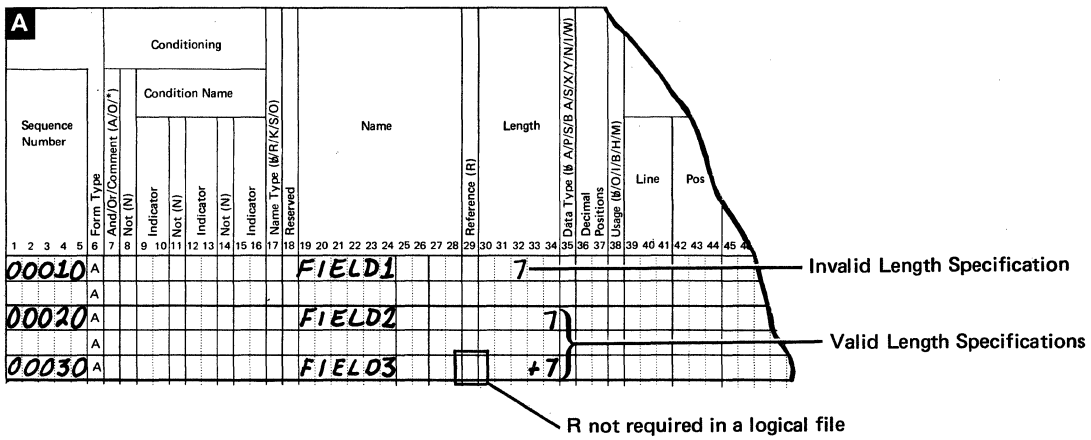
3 All records that are not associated with JSMITH in the state of New York are omitted in the first and second comparisons. Then, all records left are compared to YEAR. This way is more efficient than the ways in 1 and 2.

Position 29 (Reference)

If R is specified in position 29, the R is blanked and ignored. All logical files automatically provide the reference capability for all specified fields. Any attributes that are not specified explicitly in the logical file are furnished from the corresponding field in the physical file record format.

Positions 30-34 (Length)

If the length of the corresponding field in the associated physical file is acceptable to your program, this attribute need not be specified. If you do specify length, it must be right-justified; leading zeros are optional. The following example shows valid and invalid field length specifications:



Valid length specifications are:

Data Type	Valid Lengths
Character	1 through 32 766 characters
Binary	1 through 9 digits
Zoned decimal	1 through 31 digits
Packed decimal	1 through 31 digits

The sum of the number of bytes occupied by all fields in a record format must not exceed 32 766. The number of bytes actually occupied is determined by the system as follows:

Data Type	Bytes Occupied
Character	Number of characters
Binary	
1-4 digits	2 bytes
5-9 digits	4 bytes
Zoned decimal	Number of digits
Packed decimal	(Number of digits/2) + 1

Note: System/38 performs arithmetic operations more efficiently for packed decimal than for zoned decimal data type.

You can override the length of the field by specifying a new value or by specifying the increase or decrease in length. To increase the length, specify +n, where n is the increase. To decrease the length, specify -n, where n is the decrease. For example, an entry +4 indicates that a numeric field is to be 4 digits longer in this logical file than it is in the associated physical file. Figure 3-1 shows how to change and override the field length.

If the corresponding field in the physical file record format has a data type of binary with decimal positions greater than zero, the length cannot be overridden here. If the field you are describing is a concatenation of fields from the associated physical record format, you cannot specify the length here. The sum of the physical field lengths is used.

The field is always a fixed length. If the value of the field is numeric and fits into a smaller number of positions, it is automatically right-justified with zeros. It is left-justified with blanks if it is character data type. If the data is dropped because it is too long, no error is signaled. Figure 3-1 shows how to code the length field.

These positions are valid only for field specifications and must be blank for key, select/omit, and record format level specifications.

Note: High-level languages can restrict the field length; any such restriction should be observed for files used by these languages.

Position 35 (Data Type)

Use this position to specify the data type of the field when it is presented to your program.

The valid entries for logical files are:

Entry	Meaning
Numeric types	
P	Packed decimal
S	Zoned decimal
B	Binary
Alphabetic type	
A	Character

Figure 3-1 shows how to code the data type.

Any conversion of data types from the physical file record format is permitted within the numeric types (for example, a binary field in the physical file converted to zoned decimal in the logical file); however, if a binary field in the physical file record format has decimal positions greater than zero, the data type cannot be overridden in the logical file.

You can convert zoned decimal fields to character fields and the converse, provided that the field lengths are the same. It is your responsibility to ensure that the data is valid. For example, if the field is zoned decimal in the physical file and you specify character type (A) for presentation to your programs, you must ensure that the field contains only decimal data when it is returned through the logical file to the physical file on which it is based.

If the field in the physical file corresponding to the one you are defining has a data type that is acceptable to your programs, you need not specify the data type.

If the field you are defining is a concatenation of fields from the associated physical file (specified by the CONCAT keyword), you cannot specify the data type. It is assigned by CPF as follows:

Included Fields	Concatenated Field
Numeric (B,S,P)	Zoned decimal
Character	Character
Mixed	Character

Positions 36-37 (Decimal Positions)

If the placement of the decimal point in the corresponding field in the physical file is acceptable to your program, leave positions 36 through 37 blank.

To override or change the placement of the decimal point within a packed decimal or zoned decimal field, specify a number from 0 through 31 to indicate the number of decimal positions to the right of the decimal point. (The number here must not be greater than the number of digits specified in the field length.) Figure 3-1 shows how to specify decimal positions.

You can specify a value either to override or to change the corresponding value in the physical file. To override the value, specify the new value. To change the value, specify the amount you want the field increased or decreased and precede it with either a + or -, respectively. For example, an entry of +4 indicates there are to be 4 more digits to the right of the decimal point than there were in the corresponding field in the physical file.

If you specify a value in positions 36 through 37 and your program sends or retrieves data through the logical file field to the physical file field, CPF aligns the data on the decimal point. Depending on the case, this can cause the truncation of decimal values or a data conversion error (CPF issues CPF 5029 to your program). Decimal values are truncated in the following cases:

- When reading from a logical file that reduces the number of decimal positions specified in the physical file
- When writing to a logical file that increases the number of decimal positions specified in the physical file

For example, if the physical file field is defined as 4 digits long with 2 decimal positions and the logical file field decreases the decimal positions to 0 decimal positions, a value of 0.20 in the physical file becomes a value of 0 in the logical file, and a value of 2.52 in the physical file becomes a value of 2 in the logical file.

When decimal values are truncated, the left side of the field is filled with zeros.

A data conversion error can occur in the following cases:

- When writing to a logical file that reduces the number of decimal positions specified in the physical file
- When reading from a logical file that increases the number of decimal positions specified in the physical file

The data conversion error occurs because too many digits would be moved into the space available to the left of the decimal point. For example, if, as in the previous example, the physical file field is defined as 4 digits long with 2 decimal positions and the logical file field decreases the decimal positions to 0 decimal positions, a value of 3322 written to the logical file cannot fit in the physical file (only 2 digits are allowed left of the decimal point in the physical file).

To avoid data conversion errors, increase or decrease the length (positions 30 through 34) of the logical file field by the same amount that you increase or decrease the decimal positions.

If the field being described is a concatenation of fields from the associated physical file, you cannot specify decimal positions. (A field in the physical file that contains decimal positions cannot be included in a concatenated field.)

You cannot override decimals for binary fields (data type B). When the corresponding field in the physical file is binary and contains decimal positions greater than zero, the decimal positions in the physical file record format are used. When the logical file field is binary and the corresponding field in the physical file is not binary, the decimal positions must be zero.

Note: High-level languages can impose specific length and value restrictions on the decimal positions; these restrictions should be observed for files used by those languages.

Position 38 (Usage)

This position does not apply to logical files and must be blank unless an asterisk appears in position 7 (which indicates a comment line).

Positions 39-44 (Location)

These positions do not apply to logical files and must be blank unless an asterisk appears in position 7 (which indicates a comment line).

Keyword Entries (Positions 45-80)

This section contains keyword entries valid for defining logical files. They are entered in positions 45 through 80 (functions). See the section *Data Description Specifications Syntax Rules* in Chapter 1 for a discussion of the general rules for specifying keywords. Figure 3-2 shows how to use the general syntax rules for specifying DDS for logical files.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/060* Printed in U.S.A.

File		Keying Instruction	Graphic								Description	Page	of
Programmer	Date		Key										

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/XY/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/W)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A	*									SYNTAX FOR A LOGICAL		FILE						
00020	A	*																	
	A																		
00030	A																		KEYWORDA
	A																		
00040	A								R		RECORD								KEYWORDB
00050	A																		KEYWORDC KEYWORDD
	A																		
00060	A										FIELDA		20						KEYWORDE('This is a text example')
00070	A																		KEYWORDF(VALUEA)
00080	A																		KEYWORDG(VALUEB VALUEC)
00090	A										FIELDB		40						
00100	A										FIELDC		5	B					KEYWORDH('This text example continues with a minus sign')
00110	A																		
00120	A																		KEYWORDI('This text example +
00130	A																		continues with a plus sign')
	A																		
	A																		
	A																		
	A																		

*Number of sheets per pad may vary slightly.

Figure 3-2. Syntax for a Logical File

The following keywords are valid for logical files:

ABSVAL
ACCPTH
ALL
ALTSEQ
CHECK
CMP
COLHDG
COMP
CONCAT
DESCEND
DIGIT
EDTCDE
EDTWRD
FORMAT
LIFO
NOALTSEQ
PFILE
RANGE
RENAME
SIGNED
TEXT
UNIQUE
VALUES
ZONE

Note: When a logical file is based on a physical file to be used as a source file, you cannot use the following keywords:

ABSVAL
ALTSEQ
DESCEND
LIFO
NOALTSEQ
SIGNED
UNIQUE
ZONE

ABSVAL

Use this key field level keyword (absolute values) to specify that when CPF sequences the values associated with this numeric field, it is to ignore the sign of the field.

The following example shows six records with zoned decimal key fields:

Record	Numeric Key Field (Zoned Decimal)	Hexadecimal Representation
1	98	F9F8
2	00	F0F0
3	98-	F9D8
4	97	F9F7
5	20	F2F0
6	99	F9F9

If no sequencing keywords are specified, the records are sequenced in this order:

Record	Numeric Key Field (Zoned Decimal)	Hexadecimal Representation
2	00	F0F0
5	20	F2F0
3	98-	F9D8
4	97	F9F7
1	98	F9F8
6	99	F9F9

If the ABSVAL keyword is specified, the absolute value of the negative field is used, and the resulting sequence is:

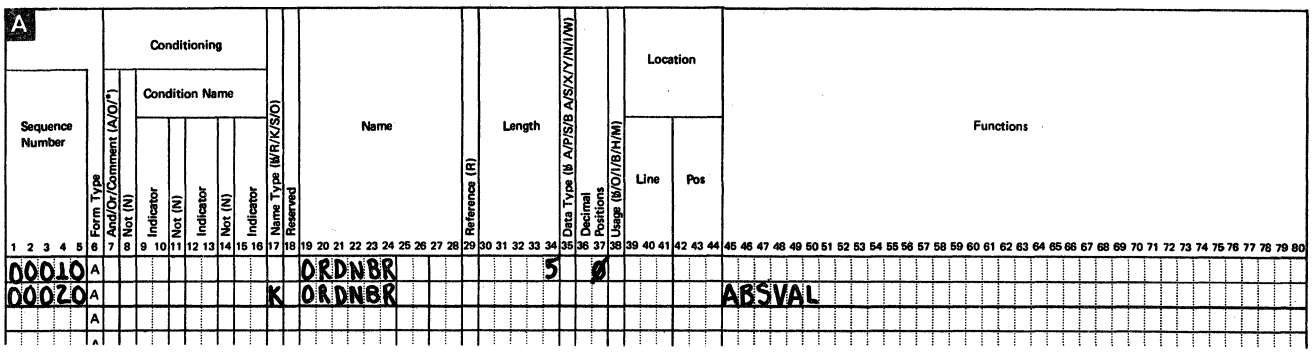
Record	Numeric Key Field (Zoned Decimal)	Hexadecimal Representation
2	00	F0F0
5	20	F2F0
4	97	F9F7
1	98	F9F8
3	98-	F9D8
6	99	F9F9

The ABSVAL keyword is not valid for a field of character data type. You cannot use it with the keywords SIGNED, ZONE, or DIGIT.

ABSVAL (a key field level keyword) and ALTSEQ (a file level keyword) are mutually exclusive. That is, if you specify ABSVAL for a key field, NOALTSEQ is in effect for that key field, even if ALTSEQ was specified at the file level (whether the NOALTSEQ keyword is specified or not).

If you do not specify SIGNED, ABSVAL, ZONE or DIGIT for a key field, the value of the field is treated as a string of unsigned binary data.

The following example shows how to specify the ABSVAL keyword:

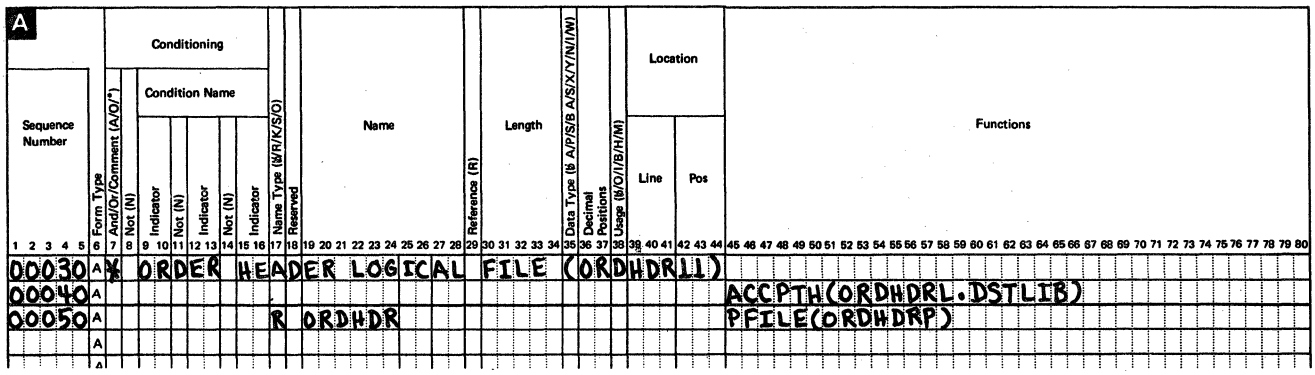


ACCPTH(data-base-file-name[.library-name])

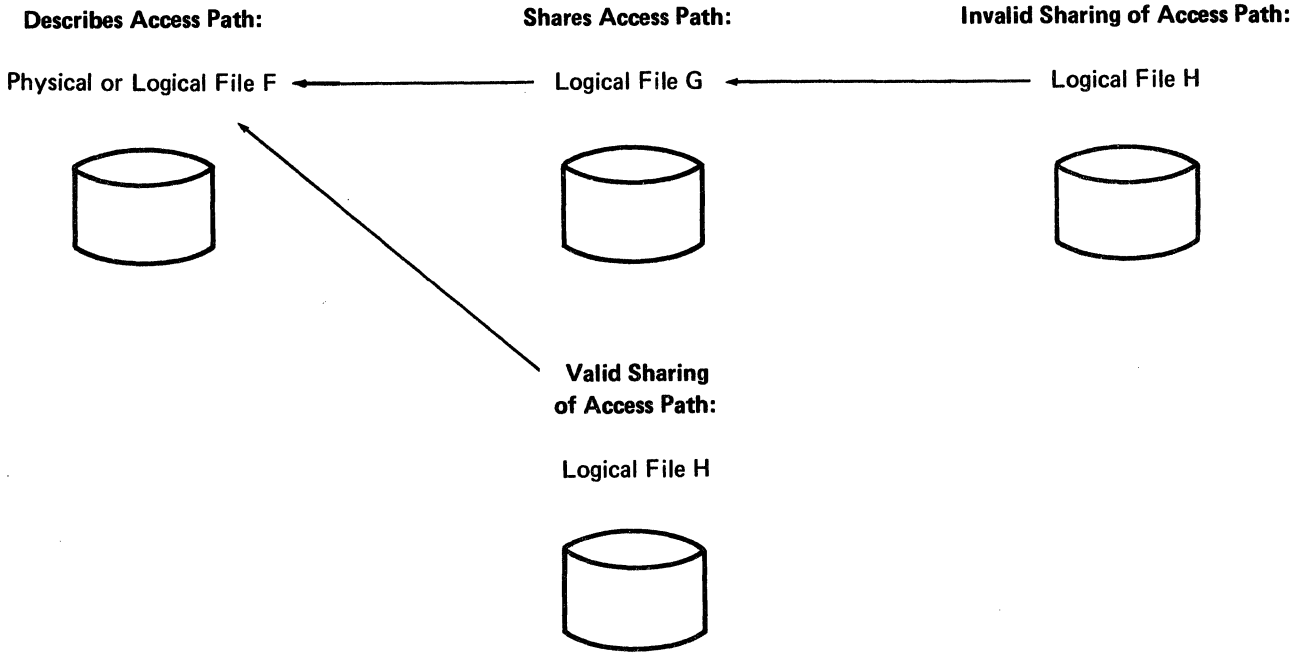
Use this file level keyword (access path) when you want this logical file to share the access path with another physical or logical file with a keyed sequence access path; the name of the file describing the access path is the parameter value for the keyword.

Use this keyword instead of specifying key fields and select/omit fields for each record format in the logical file. When you use this keyword, you can reduce the overhead required on the system for maintenance of separate access paths.

The following example shows how to specify the ACCPTH keyword:



When an access path is shared, the file describing the access path must not already be sharing an access path of another file. That is, there is only one level of access path sharing. The following illustrates how access paths *cannot* be shared.



When files share an access path, the fields in the file sharing the access path must follow the same rules that are applied to any field description for a logical file.

The file sharing the access path must specify the same physical files (specified using PFILE keywords) as the file describing the access path, and the physical files must be specified in the same order on each record format; however, the number of record formats or the allocation of physical files to the formats is variable. Also, the DTAMBRS parameter for the Create Logical File (CRTLF) command can exclude some of the members of the physical files specified on the PFILE keyword. The record formats used in the file sharing the access path must contain the same key fields and select/omit fields used in the file describing the access path.

The fields in the record formats of the file sharing the access path can be a subset of the fields in the file describing the access path, or the record format of the file sharing the access path can include more fields from the physical file record format than are specified in the file describing the access path.

ALL

Use this select/omit level keyword to specify the action to be taken after all other select/omit specifications have been processed for this logical file. Specifying ALL with S in position 17 indicates that records not meeting any of the other select/omit rules are to be selected. Specifying O in position 17 indicates that records not meeting any of the other select/omit rules are to be omitted. The ALL keyword must be the last select/omit keyword specified. You cannot specify a field name with the ALL keyword.

If the ALL keyword is not specified, the default action taken is the opposite of the last select/omit specification that you made for the file. If the last specification was a select, the default is to omit all. If the last specification was an omit, the default is to select all.

The following example shows how to specify the ALL keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								S		ACT								COMP(EQ 30000)
00020	A								S		ACT								COMP(GT 31000)
00030	A								O		AMT								COMP(LT 0)
00040	A								O										ALL
	A																		
	A																		
	A																		
	A																		

ALTSEQ(alternate-sequence-table-name[.library-name])

This file level keyword (alternate sequence) tells CPF that if a key has been specified for this file, CPF is to use an alternate collating table when sequencing the records of a file member for retrieval. The name of the alternate table is a required parameter value; the library name is optional. If the library name is not specified, the library list, *LIBL, is used.

The ALTSEQ keyword is not valid when FILETYPE(*SRC) is specified on the Create Logical File (CRTLF) command. See the *CPF Programmer's Guide* for information about the FILETYPE(*SRC) parameter value.

Notes:

1. Alternate collating sequences are not valid for key fields with a data type of either packed decimal or binary, or for key fields with SIGNED or ABSVAL specified. For those fields, NOALTSEQ (a key field level keyword) is assumed and need not be specified. NOALTSEQ can also be specified for any field in a composite key not requiring the alternate sequence.
2. At file creation time, you must have operational rights to the alternate collating table. The table is created using the Create Table (CRTTBL) command. See the *CPF Programmer's Guide* for information on how to create and enter a table.

In the following example, records with format RECORD1 are sequenced by key NAME according to the alternate collating sequence TABLE1 in library TABLELIB:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A																		ALTSEQ(TABLE1.TABLELIB)
00020	A								R		RECORD1								PFILE(PF1)
00030	A										S								
00040	A										S								
00050	A										S								
00060	A										NAME		20						
00070	A										S								
00080	A										S								
00090	A								K		NAME								
	A																		
	A																		

CHECK(edit-check-code [. . .])

This field level keyword is a validity checking keyword for use in display files. CHECK does not request any action by data base data management, but can be specified in the data base file description so that it can be duplicated into display file descriptions when this field is referenced at display file creation. The rules for specifying this keyword are the same for a data base record format as for a display file record format. See Chapter 4, *Display Files*, for information on how to specify this keyword.

The following edit/check codes are not allowed in data base files even for reference purposes:

Codes	Meanings
FE	Field exit check
RL	Right-to-left cursor movement
RLTB	Right-to-left, top-to-bottom cursor movement

CMP(relational-operator-code constant-value)

The CMP keyword is equivalent to the COMP keyword. The COMP keyword is preferred because it is compatible with DDS on other systems. See the COMP keyword description for an explanation of how to use these keywords.

COLHDG('line-1' ['line-2' ['line-3']])

This field level keyword (column heading) specifies column headings to be used as a label for this field by the query utility, the data file utility (DFU), and the screen design aid (SDA).

A maximum of three lines (of 20 characters each) is allowed.

Each line of the column heading must be enclosed in apostrophes. Use double apostrophes ('') to specify apostrophes within column headings. One or more blanks must be used to separate the first column heading line from the second and the second from the third.

If COLHDG is not specified, the column heading from the physical file is used (except when the field is a concatenation of fields, in which case the default is the field name). If COLHDG is specified and TEXT is not, 50 positions of column heading information are used as text. For example, COLHDG('Order' 'Date') is equivalent to TEXT('Order Date').

The following example shows how to specify the COLHDG keyword:

A	Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: & Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blR/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (b A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (b/O/I/B/H/M)	Location: Line	Location: Pos	Functions
	1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	0015 0	A										ORDDAT		5						COLHDG('Order' 'Date')
	0016 0	A										NAME		20						COLHDG('Customer''s Name')
	0017 0	A										CITY		20						COLHDG('Customer' 'City' 'Field')
		A																		
		A																		
		A																		
		A																		
		A																		

COMP(relational-operator-code [value | field-name])

At the select/omit level, use this keyword (comparison) to specify that CPF is to make a comparison using the value of the field you are defining. CPF compares this field with the value of another field within the same record or compares this field with a constant value that you specify. The relational operator code determines whether the results of the comparison meet the criteria for omission/selection.

At the field level, use this keyword as a validity checking keyword that can be specified in a logical file so that it can be duplicated into a display file description through the reference function. (See Chapter 4, *Display Files*, for information on how to specify this keyword for validity checking.)

The valid relational operator codes are:

Relational Operator Code	Meaning
EQ	Equal
NE	Not equal
LT	Less than
NL	Not less than
GT	Greater than
NG	Not greater than
LE	Less than or equal
GE	Greater than or equal

At the select/omit level, a field name can be specified as a parameter value for this keyword. The field name must be specified within the physical file record format (that is, not renamed or a concatenated logical name). A field name cannot be specified when COMP is used as a validity checking keyword. The parameter values must correspond to the data type and the length of the field specified. Character parameter values must be enclosed in apostrophes; numeric parameter values must not be enclosed in apostrophes.

Note: If the value specified is a constant and the field named (to be compared to) is a numeric field, alignment is based on the decimal precision specified, and leading and trailing blanks are filled with zeros. If no decimal point is specified, the decimal point is assumed to be to the right of the last (rightmost) digit. For example, for a numeric field with a length of 5 (specified in column 34) and 2 decimal positions (specified in column 37), 1.2 is interpreted as 001.20, and 100 is interpreted as 100.00.

The following example shows how to specify the COMP keyword as a select/omit keyword:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B/A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Pos	Functions
1 2 3 4 5	6	7	9	10 11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	
00010	A							S		ACT								COMP(GE 'A')	
00020	A							S		AMT								COMP(NE 0)	
00030	A									AMT								COMP(EQ FIELDA)	
	A																		
	A																		
	A																		

CONCAT(field-1 field-2...)

Use this field level keyword (concatenate) when you want to combine two or more fields from the physical file record format into one field in the logical file record format you are describing. The name of this concatenated field must appear in positions 19 through 28. Specify the physical file field names in the order in which you want them to be concatenated, and separate them by blanks.

Note: The sequence in which the field names are specified in the logical file is important. If the same physical field is specified more than once in a record format in the logical file (that is, by using either RENAME or CONCAT), the sequence in which the fields are specified in the logical file is the sequence in which the data is moved to the physical file. Thus, the value of the field the last time the field is specified in the logical file is the value in the physical record. If a renamed (RENAME keyword) or concatenated (CONCAT keyword) field is used as a key field, the value of the key for the purposes of ordering the record is the value assigned to the first occurrence of the field in the logical file record format.

The following rules and restrictions apply:

- A field containing decimal positions other than zero *cannot* be included in a concatenated field. A field having decimal positions of zero *can* be included, in which case the field is treated as an integer field.

- The length of the concatenated field is the sum of the lengths (digits and characters) of the fields that are included in the concatenation.

- Data type is assigned by CPF as follows:

Fields Being Concatenated	Concatenated Field
Numeric (B,S,P)	Zoned decimal (S)
Character (A)	Character (A)
Alphameric	Character (A)

- When concatenating numeric fields, the sign of the rightmost field in the concatenation is used as the sign of the concatenation. The signs of the other fields are ignored; however, they are present in the concatenation. Therefore, if a negative value can appear in a field other than the last, you must take appropriate action to eliminate the embedded signs (such as converting the concatenated field to packed decimal).

- The maximum length of a concatenated field varies depending on the data type of the concatenated field and the length of the fields being concatenated. If the concatenated field is zoned decimal (S), its total length cannot exceed 31 bytes; if it is character (A), its total length cannot exceed 32 766 bytes.

The following examples show how to specify the CONCAT keyword:

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00010	A								R		RECORD1								PFILE(PF1)
00020	A										DATE								CONCAT(MTH DAY YEAR)
	A																		
	A																		
	A																		

In this example, MTH, DAY, and YEAR are fields in the physical file that are concatenated into one field DATE in the logical file.

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00010	A								R		RECORD2								PFILE(PF1)
00020	A										DATE								CONCAT(MTH DAY YEAR)
00030	A										MTH								
00040	A										DAY								
00050	A										YEAR								
	A																		
	A																		
	A																		

In this example, if the program changes DATE from 01 03 81 to 02 05 81, the value placed in the physical record does not change because the fields specified last are MTH (value 01), DAY (value 03), and YEAR (value 81); however, if MTH, DAY, and YEAR are changed to new values, the value of DATE in the physical record also changes.

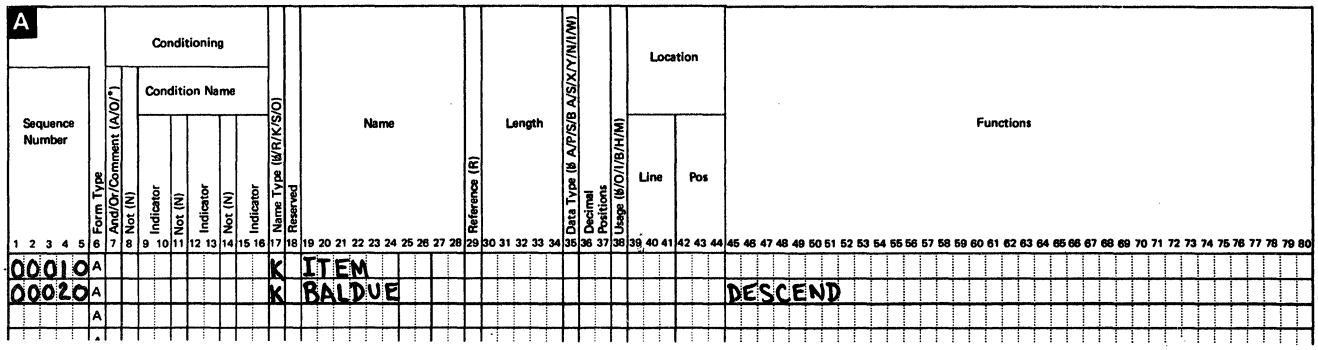
Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00010	A								R		RECORD3								PFILE(PF1)
00020	A										DATE								CONCAT(MTH DAY YEAR)
00030	A										CMPDAT								CONCAT(DAY MTH YEAR)
	A																		
	A																		
	A																		

In this example, fields from the physical file are concatenated into more than one field in the logical file.

DESCEND

Use this key field level keyword to specify that the values of this character or numeric key field are to be retrieved in descending sequence. If this keyword is not specified, ascending sequence is assumed.

The following example shows how to specify the DESCEND keyword:



DIGIT

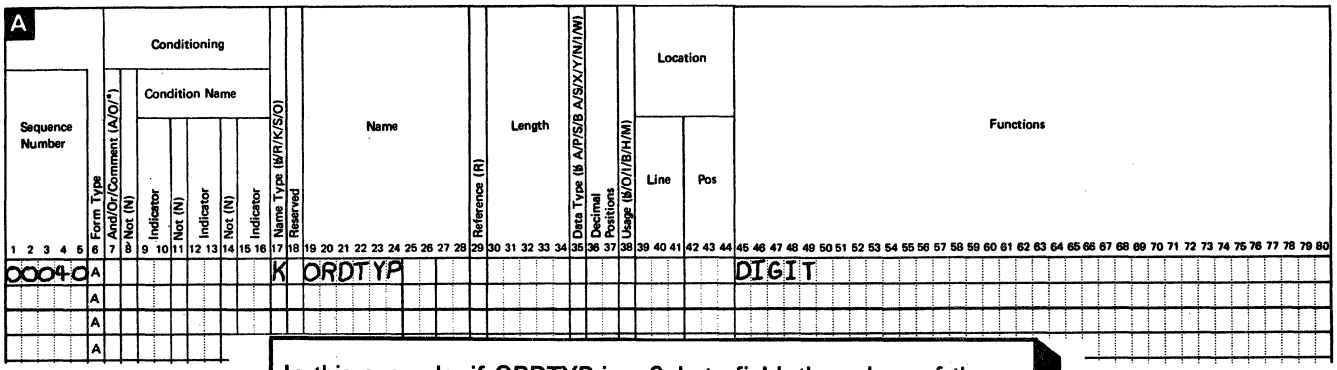
This key field level keyword specifies that only the digit portion (rightmost 4 bits) of each byte of the key field is to be used when constructing a value associated with this key field. The zone portion is filled with zeros.

This keyword is applied against the entire key field (not just a position within the field); it is valid only for character or zoned decimal type fields.

You cannot use this keyword with the ABSVAL, the SIGNED, or the ZONE keyword.

If you do not specify SIGNED, ABSVAL, ZONE, or DIGIT for a key field, the value of the field is treated as a string of unsigned binary data.

The following example shows how to specify the DIGIT keyword:



In this example, if ORDTYP is a 3-byte field, the values of the field for three different records could be as follows:

Values	Hexadecimal	Digits Used for Key
C4J	C3F4D1	341
LMA	D3D4C1	341
3D1	F3C4F1	341

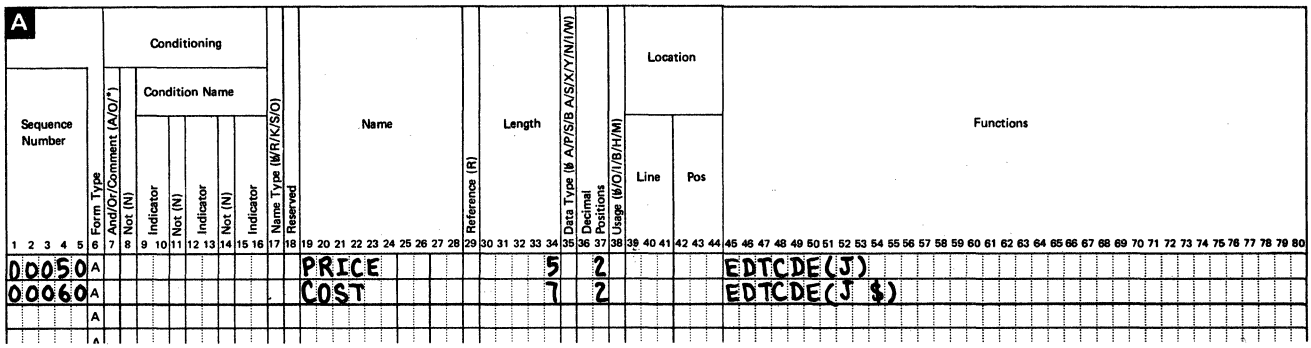
EDTCDE(edit-code [* | floating-currency-symbol])

Use this field level keyword to specify an edit code for editing of output data using this current field description. The edit code specification is not used by data base data management, but is retained in the data base file description so it can be duplicated into device file descriptions through the reference function at device file creation. An edit code is a required parameter value for this keyword.

EDTCDE is valid for numeric fields only. You cannot specify both the EDTCDE and EDTWRD keywords for the same field. You can use the EDTCDE keyword to perform edit functions, such as the following:

- Suppress leading zeros
- Omit a sign from the low-order position of the field
- Punctuate the field without having to specify an edit word (see the EDTWRD keyword)
- Perform asterisk fill
- Specify floating currency symbol
- Perform user-defined edit functions

The following example shows how to specify the EDTCDE keyword:



A	Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/(N)/I/W)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Location: Pos	Functions
	1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	00050	A										PRICE		5		2				EDTCDE(J)
	00060	A										COST		7		2				EDTCDE(J $)
		A																		
		A																		

CPF Edit Codes: The CPF edit codes that you can specify as a parameter value for this keyword are:

1 through 4
A through D
J through M
X through Z

See the EDTCDE keyword descriptions in Chapter 4, *Display Files*, and Chapter 5, *Printer Files*, for summaries of the functions provided by all CPF edit codes except X. Edit code X is used only to override the default edit codes used in query (default edit code is J) and the data file utility (default edit code is L).

User-Defined Edit Codes: You can also specify any of five user-defined edit codes (5 through 9). Before creating a device file from DDS source containing user-defined edit codes, the user-defined edit codes must exist. When creating the device file, the editing information is extracted from the user-defined edit codes as referenced in the DDS. Changing user-defined edit codes does not affect existing device files unless the device files are re-created. See the *CPF Programmer's Guide* and the *CL Reference Manual* for a description of how to create user-defined edit codes if those shipped with the system cannot be used.

Asterisk Fill and Floating Currency Symbol: You can specify asterisk fill or floating currency symbol with edit codes 1 through 4, A through D, and J through M.

When you specify asterisk fill, an asterisk (*) is written for each zero suppressed. A complete field of asterisks is printed for a zero balance field.

When you specify a floating currency symbol, the symbol appears to the left of the first significant digit. The symbol does not print on a zero balance when an edit code is used that suppresses the zero balance. The symbol that you specify must match the system value for the floating currency symbol (QCURSYM) when the file is created; it does not have to match when the file is used. See the *CPF Programmer's Guide* for a description of how to change this symbol.

See the EDTCDE keyword descriptions in Chapter 4, *Display Files*, and Chapter 5, *Printer Files*, for examples of valid edit codes, source data, and edited output.

EDTWRD('edit-word')

If you cannot accomplish the desired editing by using the EDTCDE keyword, you can specify an edit word instead. The edit word specification is not used by data base data management, but is retained in the data base file description so that it can be duplicated into device file descriptions through the reference function at device file creation. The edit word specifies the form in which the field values are to be displayed and clarifies the data by inserting characters directly (such as decimal points, commas, floating and fixed currency symbols, and credit balance indicators). EDTWRD can also be used to suppress leading zeros and provide asterisk-fill protection.

The following rules apply when you specify an edit word:

- The edit word must be enclosed in apostrophes.
- The EDTWRD keyword is valid for numeric fields only.
- The number of replaceable characters in the edit word must equal the length of the field.
- When you are using the floating currency symbol, the sum of the number of blanks and the stop-zero-suppression character (digit positions) contained in the edit word must be equal to the number of positions in the field to be edited. The currency symbol is not counted as a digit position.

  For example, if you specify a field length of 7 with 2 decimal positions, the edit word must be specified as:

  EDTWRD('b b b b$0.b b')

  where the b represents a blank.
- Any printable character is valid, but the following characters in certain positions have special meanings:
  - *Blank*: A blank is replaced with the character from the corresponding position of the data field. A blank position is referred to as a digit position.
  - *Ampersand*: An ampersand causes a blank in the edited field. The ampersand is not printed.
  - *Zero*: A zero stops zero suppression; place it in the rightmost position where zero suppression is to stop. The zero is replaced with the character from the corresponding position of the data field, unless that character is zero. Any zeros in the data that appear to the right of the stop-zero-suppression character are printed. At least one leading zero is suppressed. Each zero that is suppressed is replaced by a blank.

- *Asterisk*: An asterisk stops zero suppression and replaces zeros with asterisks (asterisk protection). Place the asterisk in the rightmost position where zero suppression is to stop. Each zero that is suppressed is replaced by an asterisk. An asterisk preceding a zero is interpreted as representing asterisk protection. In this case, the zero prints as a constant.

  An asterisk is considered a digit replace position.

- *Currency Symbol*: A currency symbol coded immediately to the left of the zero suppression code causes the insertion of a currency symbol in the position to the left of the first significant digit. It is called the *floating currency symbol* when used in this manner.

  A currency symbol coded in the leftmost position of the edit word is fixed and prints in the same location each time. When used in this manner, it is called the *fixed currency symbol*.

  The currency symbol is not considered a digit replace position. This symbol must correspond to the QCURSYM system value.

- *Decimals and Commas*: Decimals and commas are printed in the same relative positions in which they are coded in the edit word unless they are to the left of the first significant digit. In that case, the positions are filled in with blanks or replaced by asterisks.

- *All Other Characters*: All other characters are printed if they are to the right of the significant digits in the data field. If they are to the left of the high-order significant digits in the data, these positions are filled with blanks or replaced by asterisks if asterisk protection is being used.

- If you want to show a negative sign with a negative number, include a sign in the edit word. Use either the minus sign (-) or the letters CR (credit). These print only if the number is negative.

- You cannot specify both the EDTWRD and EDTCDE keywords for the same field.

See the EDTWRD keyword description in Chapter 4, *Display Files*, for examples of edit words, source data and edited output.

The following example shows how to specify the EDTWRD keyword:

```
00010A            CRYCST         7  2       EDTWRD('    $Ø.  ')
     A
     A
```

FORMAT(data-base-file-name[.library-name])

Use this record level keyword to specify that this record format is to share the specifications for a previously defined record format. The name of the record format you are defining must be the name of the previously defined record format. The data base file name is required; it is the name of the physical or logical file from which the previously defined record format is taken. The library name is optional; if you do not specify it, the library list in effect at file creation time is used.

If you specify the FORMAT keyword, you cannot specify field specifications for this record format. However, you must specify a key and, if necessary, select/omit specifications if you want them to be in effect for this file.

If the data base file from which you are using the record format is deleted, the record format remains in existence as long as some file is using the record format. For example, a record format RECORD in FILE2 uses the FORMAT keyword to share the specifications of an existing record format RECORD in FILE1. Both files have been created. If FILE1 is deleted and then re-created with different DDS, RECORD still exists in FILE2 and can itself be referenced for the original record format by other files using the FORMAT keyword.

The following example shows how to specify the FORMAT keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		CUSREC								PFILE(CUSMSTP)
00020	A																		FORMAT(CUSMSTL)
	A																		
	A																		
	A																		

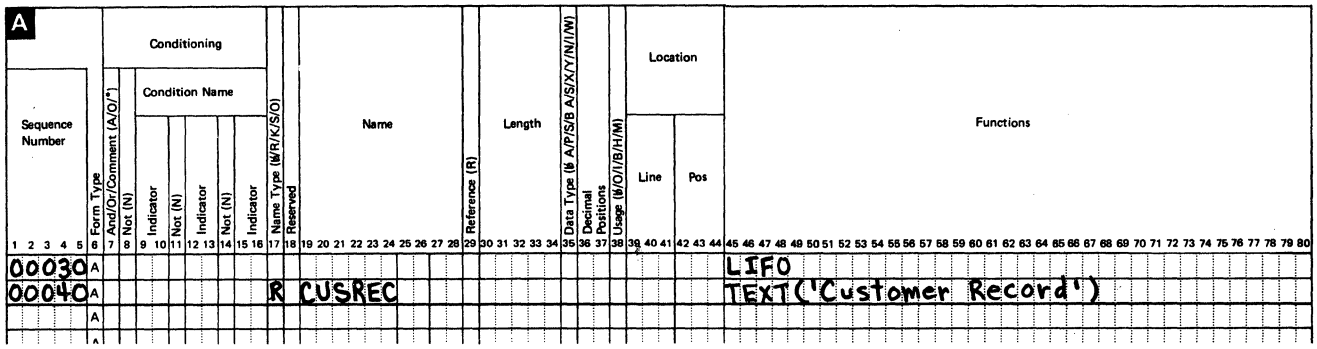
The record format for this physical file is the same as the previously specified record format in file CUSMSTL. The name of this record format (CUSREC) must be the same as the name of the record format in CUSMSTL.

LIFO

Use this file level keyword to specify that if records with duplicate key values are retrieved from one based-on physical file member, they are to be retrieved in a last-in-first-out (LIFO) order.

If neither LIFO nor UNIQUE is specified, records that have duplicate key values are retrieved in a first-in-first-out (FIFO) order.

The following example shows how to specify the LIFO keyword:



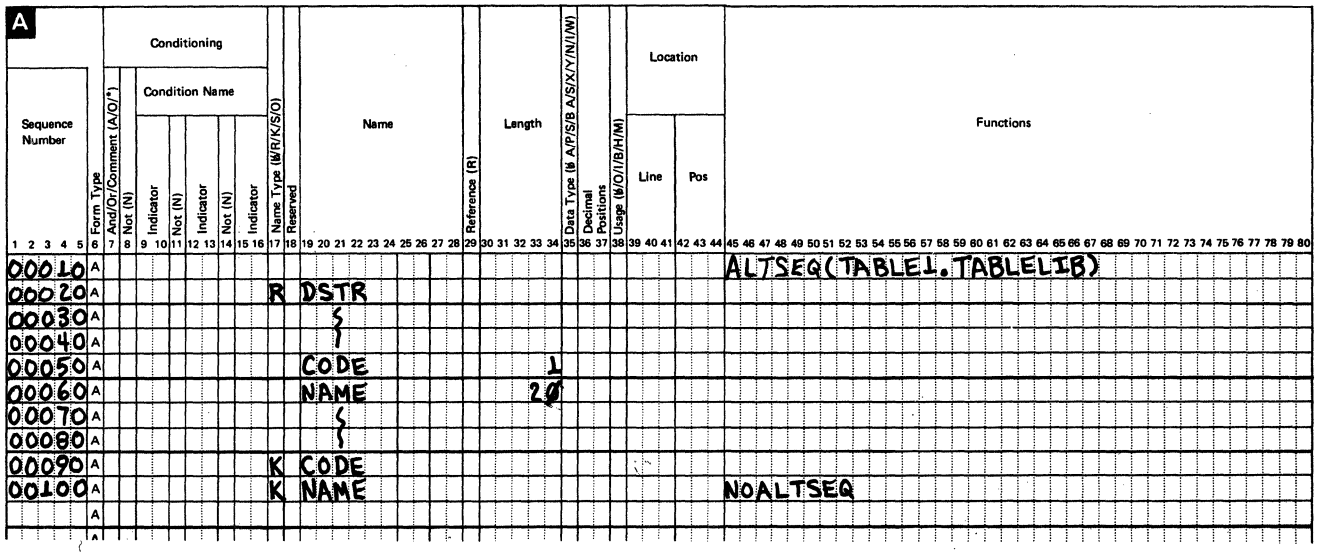
Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00030	A																		LIFO
00040	A								R		CUSREC								TEXT('Customer Record')
	A																		

The LIFO keyword is not valid when FILETYPE(*SRC) is specified on the Create Logical File (CRTLF) command. See the *CPF Programmer's Guide* for information about the FILETYPE(*SRC) parameter value.

NOALTSEQ

Use this key field level keyword to specify that the ALTSEQ keyword specified at the file level is not to apply to this key field. If you specify ABSVAL or SIGNED for a key field, NOALTSEQ is in effect whether or not the NOALTSEQ keyword is specified for that key field.

The following example shows how to specify the NOALTSEQ keyword:



Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A																		ALTSEQ(TABLE1.TABLELIB)
00020	A								R		DSTR								
00030	A										S								
00040	A																		
00050	A										CODE		1						
00060	A										NAME		20						
00070	A										S								
00080	A																		
00090	A								K		CODE								
00100	A								K		NAME								NOALTSEQ
	A																		

PFILE(physical-file-name[.library-name][.32])

Use this record level keyword to identify the physical file(s) containing the data to be accessed through the record format you are now defining. The PFILE keyword is required on every record format in a logical file. Up to 32 physical file names can be specified on PFILE keywords in a logical file. If the maximum is being used, 32 physical file names can be specified on one record format (using one PFILE keyword) or 32 physical file names can be distributed among 32 record formats; or, file names can be unevenly distributed among record formats. In any case, the maximum number of physical file names allowed is 32.

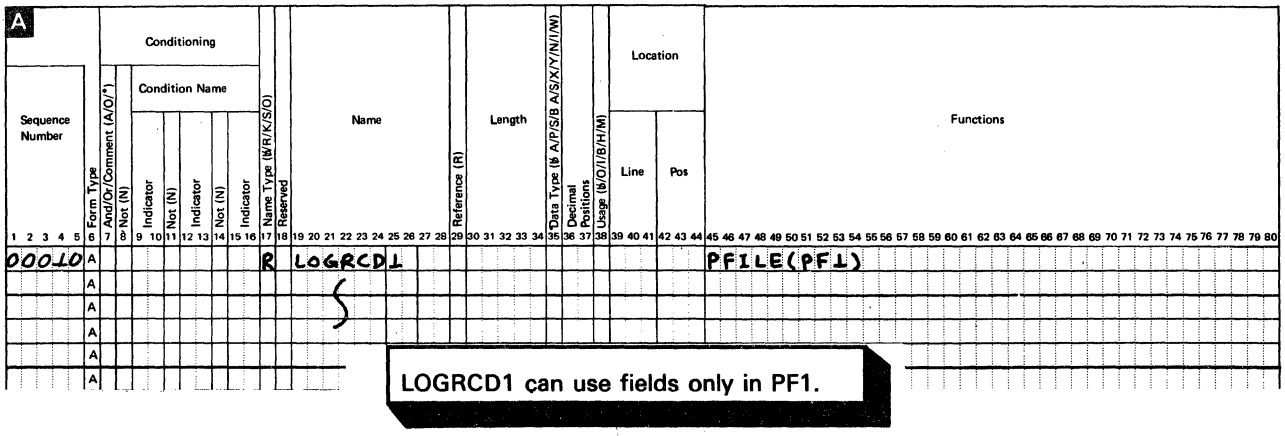
For each physical file name, a library name is optional. If the library name is omitted, the library list at file creation time is used.

If more than one physical file name is specified for one record format in the logical file, all fields in the record format for the logical file must exist in all physical files specified. If your program requires access to fields that occur in one or more of the based-on physical files, but not in all of them, you can specify a separate logical file record format that includes those fields. For instance, if FLD1 and FLD2 occur in physical files PF1, PF2, and PF3, but FLD3 occurs only in PF3, you cannot specify FLD3 in a logical file record format based on PF1 and PF2. To provide access to FLD3, specify a second logical file record format that includes FLD3. This means that *you cannot use a logical file to bring together into one record format fields from separate physical files.* A record read through the use of a record format in a logical file can contain data from only one physical file, and a record written through the use of a record format in a logical file can be stored only in one physical file.

If this logical file shares an access path defined by another physical or logical file (the ACCPTH keyword is specified for this file), then the physical files specified with the PFILE keywords in this file must be the same as the physical file specified on the other file, and they must be specified in the same order.

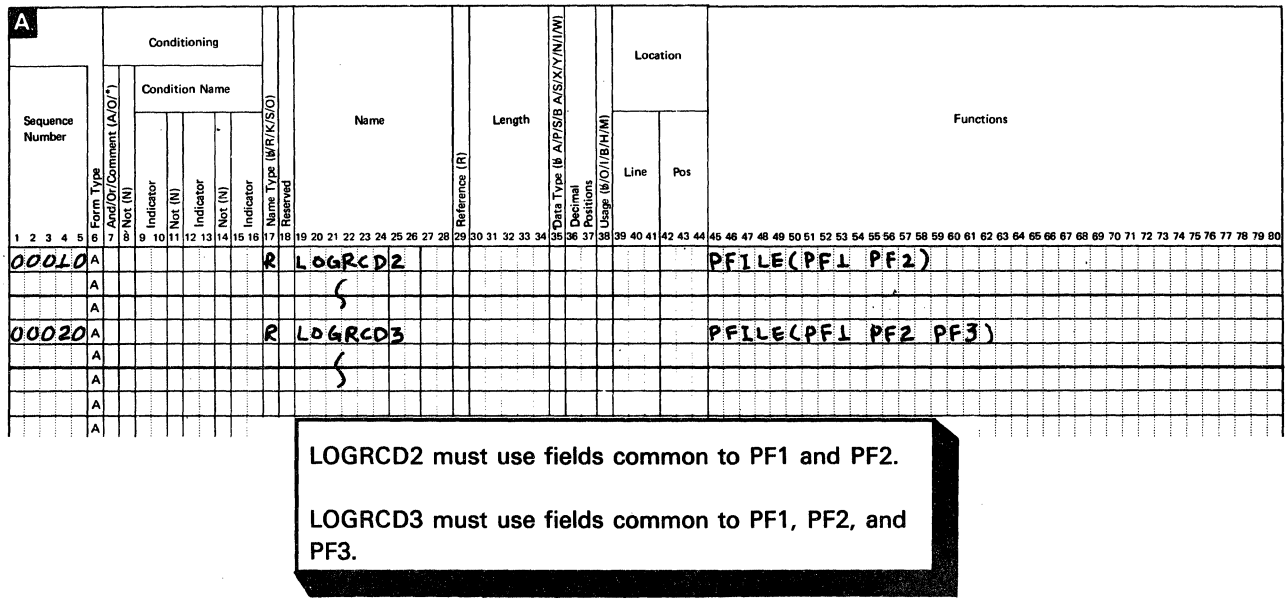
Members of physical files can be specified on the DTAMBRS parameter on the Create Logical File (CRTLF) or Add Logical File Member (ADDLFM) commands. All such physical files must be specified on the PFILE keyword at file creation time; otherwise, the commands do not complete successfully.

The following examples show how to specify the PFILE keyword:



Sequence Number	Form Type	And/Or/Comment (A/O/*)	Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blR/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (bl A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (bl O/I/B/H/M)	Location Line	Location Pos	Functions
00010	A								R		LOGRCD1								PFILE(PF1)
	A																		
	A																		
	A																		
	A																		
	A																		

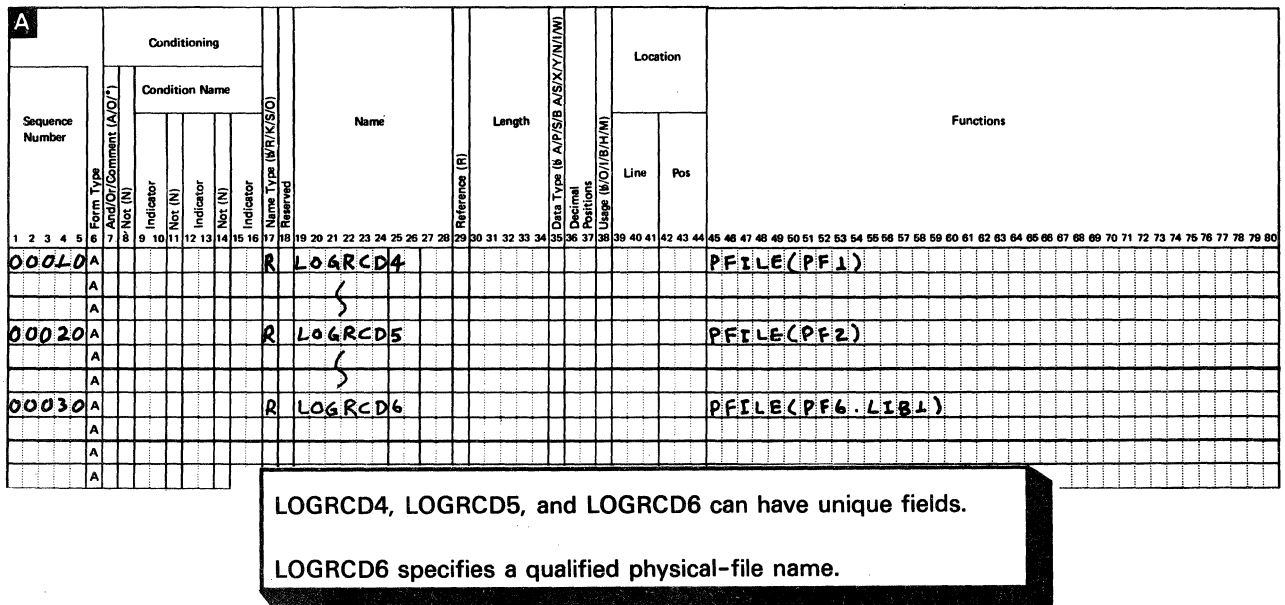
LOGRCD1 can use fields only in PF1.



Sequence Number	Form Type	And/Or/Comment (A/O/*)	Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blR/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (bl A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (bl O/I/B/H/M)	Location Line	Location Pos	Functions
00010	A								R		LOGRCD2								PFILE(PF1 PF2)
	A																		
	A																		
00020	A								R		LOGRCD3								PFILE(PF1 PF2 PF3)
	A																		
	A																		
	A																		
	A																		

LOGRCD2 must use fields common to PF1 and PF2.

LOGRCD3 must use fields common to PF1, PF2, and PF3.



Sequence Number	Form Type	And/Or/Comment (A/O/*)	Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blR/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (bl A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (bl O/I/B/H/M)	Location Line	Location Pos	Functions
00010	A								R		LOGRCD4								PFILE(PF1)
	A																		
	A																		
00020	A								R		LOGRCD5								PFILE(PF2)
	A																		
	A																		
00030	A								R		LOGRCD6								PFILE(PF6.LIB1)
	A																		
	A																		
	A																		

LOGRCD4, LOGRCD5, and LOGRCD6 can have unique fields.

LOGRCD6 specifies a qualified physical-file name.

RANGE(low high)

At the select/omit level, use this keyword to specify a range of values that are used to determine which records from the associated physical file are to be selected or omitted for the logical file. Specify the lower limit first, followed by one or more blanks, and then the higher limit.

At the field level, use this keyword as a validity checking keyword for display files. In this case, RANGE is not used by data base data management. It is specified in the logical file description so that it can be duplicated into display file descriptions when this field is referenced at display file creation. (See Chapter 4, *Display Files*.)

Note: If the value specified is a constant and the field named (to be compared with) is a numeric field, alignment is based on the decimal precision specified, and the leading and trailing blanks are filled with zeros. If no decimal point is specified, the decimal point is assumed to be to the right of the last (rightmost) digit. For example, for a numeric field with a length of 5 (specified in column 34) and 2 decimal positions (specified in column 37), 1.2 is interpreted as 001.20, and 100 is interpreted as 100.00.

The following example shows how to specify the RANGE keyword:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		RECORD								PFILE(PF1)
	A																		
00020	A										FIELDA		1		0				RANGE(2 5) 1
00030	A										FIELDB		1						RANGE('2' '5')
00040	A										FIELDC								
00050	A								K		FIELDD								
00060	A								S		FIELDA								RANGE(1 3) 2
	A																		
	A																		
	A																		

1 RANGE is specified for FIELDA as a validity checking keyword for display files that reference FIELDA. Only values 2 through 5 can be entered by a work station user.

2 RANGE is specified for FIELDA as a select/omit level keyword. FIELDA is selected in a logical file only if it contains a value between 1 and 3.

RENAME(physical-file-field-name)

Use this field level keyword when you want a field name in the logical record format you are defining to be different from its corresponding physical file field name. The name as it appears in the physical file record format is the parameter value for this keyword. One field in the physical file record format can be renamed to more than one field in the record format being described.

The following are three situations in which you might want to rename fields:

- Programs were written using a different name for the same field.
- You want to map one field in a physical file record format to two or more fields in a logical file record format.
- You are using a language that does not permit two fields having different names to have only one data storage area. (RPG III is such a language.) By specifying the RENAME keyword, you allow both fields to access the same data storage area.

Note: The sequence in which the field names are specified in the logical file is important. If the same physcial field is specified more than once in a record format in the logical file (that is, by using either RENAME or CONCAT), the sequence in which the fields are specified in the logical file is the sequence in which the data is moved to the physical file. Thus, the value of the field the last time the field is specified in the logical file is the value in the physical record. If a renamed (RENAME keyword) or concatenated (CONCAT keyword) field is used as a key field, the value of the key for the purposes of ordering the record is the value assigned to the first occurrence of the field in the logical file record format.

The following example shows how to specify the RENAME keyword:

A

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Conditioning: Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00050	A										ITMNBR								
00060	A										SHDATE								CONCAT(SHPMO SHPDA SHPYR)
00070	A										QTY								RENAME(QTYDUE)
	A																		
	A																		

SIGNED

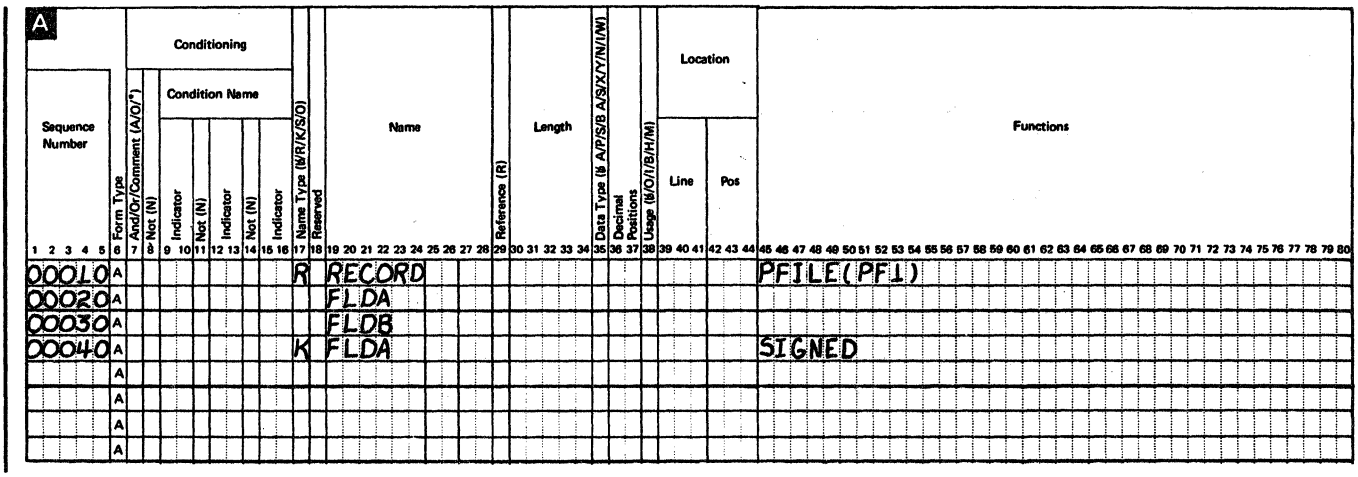
This key field level keyword specifies that when sequencing the values associated with this numeric key field, CPF is to consider the signs of the values (negative versus positive values).

This keyword is not valid for a field of character data type. You cannot use it with the ABSVAL, the ZONE, or the DIGIT keyword.

SIGNED (a key field level keyword) and ALTSEQ (a file level keyword) are mutually exclusive. If you specify SIGNED for a key field, NOALTSEQ is in effect for that key field even if ALTSEQ was specified at the file level (whether or not the NOALTSEQ keyword is specified).

If you do not specify SIGNED, ABSVAL, ZONE, or DIGIT for a key field, the value of the key is treated as a string of unsigned binary data.

The following example shows how to specify the SIGNED keyword:

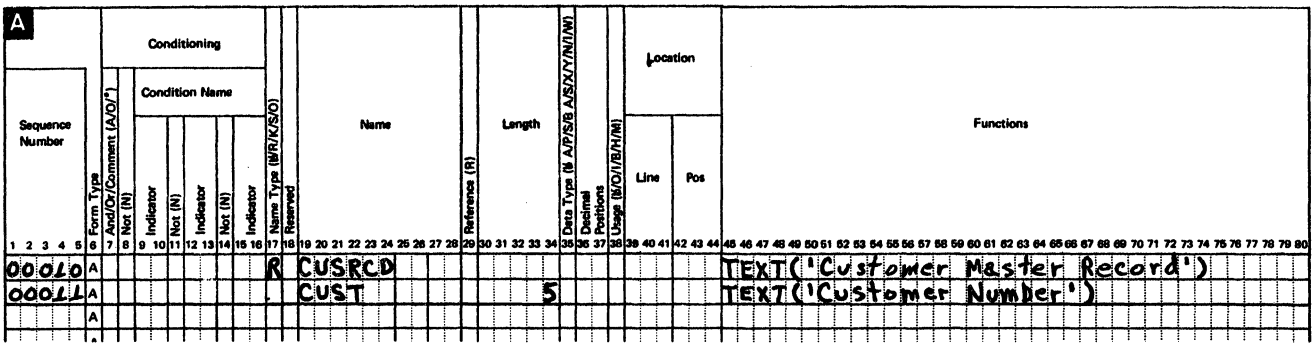


A	Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blR/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blA/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (bl/O/I/B/H/M)	Location: Line	Pos	Functions
	1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	00010	A								R		RECORD								PFILE(PF1)
	00020	A										FLDA								
	00030	A										FLDB								
	00040	A								K		FLDA								SIGNED
		A																		
		A																		
		A																		
		A																		

TEXT('description')

Use this record or field level keyword to supply a text description. The description must be enclosed in apostrophes, with a maximum of 50 characters. If more than 50 characters are specified, the keyword is accepted; however, the excess characters are dropped and a warning message is issued.

The following example shows how to specify the TEXT keyword:



A	Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blR/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blA/P/S/B A/S/X/Y/N/I/N/I/N)	Decimal Positions	Usage (blO/I/B/H/N)	Location: Line	Pos	Functions
	1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	00010	A								R		CUSRCD								TEXT('Customer Master Record')
	00011	A										CUST		5						TEXT('Customer Number')
		A																		

If TEXT is not specified and the COLHDG keyword is specified, 50 positions of column heading information are used as a default for TEXT.

If COLHDG is not specified and the field is not concatenated, the text is taken from the physical field text if it exists. If COLHDG is not specified and the field is concatenated, there is no text.

Note: You can specify TEXT at both the record format level *and* the field level. Both TEXT and FORMAT cannot be specified at the record format level (the format already contains text).

UNIQUE

Use this keyword at the file level to specify that records with duplicate key values are not allowed within a member of this logical file. Any inserts or additions of new records, or updates to existing records, that would result in a duplicate key are rejected. The application program issuing the write or the update operation receives a message. When a work station user is using DFU, a message is displayed at the work station. A copy file command copying records that would have duplicate keys in this file cannot be completed.

When the UNIQUE keyword is specified for a logical file, records in any of the based-on physical files cannot have duplicate key values if the duplicate key values occur because of keys specified in the logical file. Also, when the UNIQUE keyword is specified for a logical file, the MAINT(*IMMED) parameter value must be specified on the Create Logical File (CRTLF) command that creates the file. This means that the access path is maintained immediately as changes are made. See the *CPF Programmer's Guide* for a description of the MAINT(*IMMED) parameter value.

If the UNIQUE keyword is not specified, records with duplicate key values are sequenced in first-in-first-out (FIFO) order, unless the LIFO keyword is specified, in which case they are sequenced in last-in-first-out order. For an explanation of how records with duplicate key values are sequenced when those records are not in the same file member, see the section on data base records in the *CPF Programmer's Guide*.

The following example shows how to specify the UNIQUE keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/N)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A	*																	
00020	A	*		SAMPLE LOGICAL FILE (CUSMSTL)															
00030	A	*																	
00040	A																		UNIQUE
00050	A								R		CUSREC								PFILE(CUSMSTP)
00060	A																		TEXT('Logical File Master Record')
00070	A										CUST								
00080	A										NAME								
00090	A										ADDR								
00100	A								K		CUST								
	A																		
	A																		
	A																		

VALUES(value-1 [value-2...[value-20]])

Use this keyword at the select/omit level to specify a list of up to 20 values. The values are used to determine which records from the associated physical file are selected or omitted for this logical file. If a select/omit field key value is equal to one of the values, the record containing that value is selected or omitted.

You can also use this keyword at the field level as a validity checking keyword for display files. Specify this keyword on a field not within a select/omit specification. In this case, VALUES is not used by data base data management. Instead, it is retained so that it can be duplicated into display file descriptions through the reference function at display file creation. (See Chapter 4, *Display Files*.)

Note: If the value specified is a constant and the field named (to be compared with) is a numeric field, alignment is based on the decimal precision specified, and the leading and trailing blanks are filled with zeros. If no decimal point is specified, the decimal point is assumed to be to the right of the last (rightmost) digit. For example, for a numeric field with a length of 5 (specified in column 34) and 2 decimal positions (specified in column 37), 1.2 is interpreted as 001.20, and 100 is interpreted as 100.00.

The following example shows how to specify the VALUES keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		RECORD1								PFILE(PF1)
00020	A										FIELDA		1		0				VALUES(1 6 9)
00030	A										FIELDB		1						VALUES('A' 'B' 'C')
00040	A								K		FIELDA								
00050	A								S		FIELDA								VALUES('A' 'B' 'C')
	A																		
	A																		
	A																		

ZONE

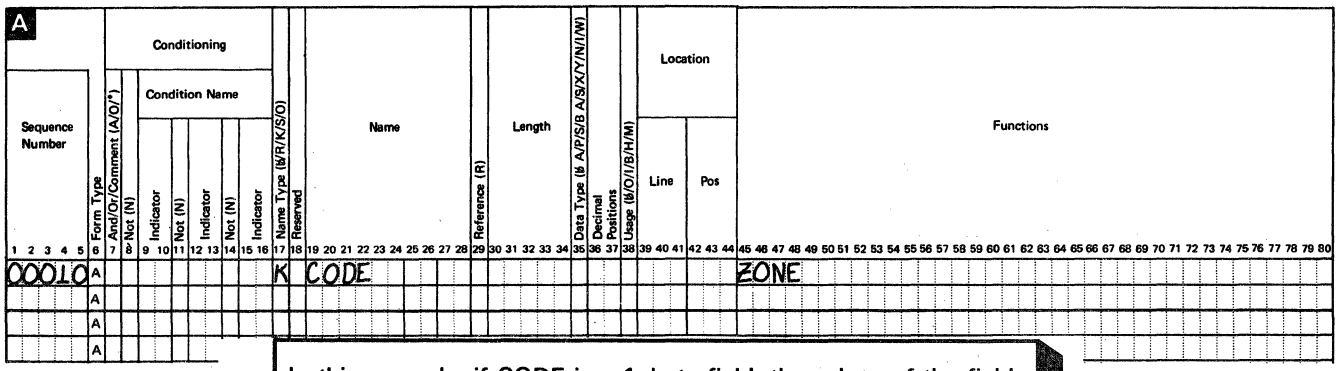
Use this key field level keyword to specify that only the zone portion (leftmost 4 bits) of each byte of the key field is to be used when constructing a value to be associated with this key field. The digit portion is filled with zeros.

This keyword is applied against the entire key field (not just a position within the field) and it is valid only for character or zoned decimal type fields.

You cannot use ZONE with the ABSVAL, the SIGNED, or the DIGIT keyword.

If you do not specify SIGNED, ABSVAL, ZONE, or DIGIT for a key field, the value of the field is treated as a string of unsigned binary data.

The following example shows how to specify the ZONE keyword:



In this example, if CODE is a 1-byte field, the values of the field for three different records could be as follows:

Values	Hexadecimal	Zone Used for Key
A	C0	C
B	C1	C
E	C4	C

Chapter 4. Display Files

This chapter is divided into two sections. The first, *Positional Entries*, provides rules and examples for filling in positions 1 through 44 of the Data Description Specifications form.

The second section of this chapter, *Keyword Entries*, provides rules and examples for specifying DDS keywords. The keywords are described in alphabetical order.

For guidance in choosing positional entries and keywords for display files, see the *CPF Programmer's Guide*.

DEFINING A DISPLAY FILE

To define a display file, specify the entries in the following order:

1. File level entries (optional)
2. Record level entries
3. Field level entries (optional)

} Repeat these entries for each record format in the file.

Specify at least one record format in the file. The maximum number of record formats in a display file is 1024. The maximum number of fields in any one record format is 32 767. The maximum combined length of all named fields and indicators in a record format is 32 767, regardless of usage (I, O, B, M, H).

Note: The file name is specified through the Create Display File (CRTDSPF) command, not through DDS.

Complete display file examples can be found in Appendix F, *Examples*.

An explanation of *file level*, *record level*, and *field level* can be found in Chapter 1, *Introduction*.

The syntax rules for specifying DDS keywords can be found in Chapter 1, *Introduction*.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050* Printed in U.S.A.

File		Keying Instruction	Graphic								Description	Page	of
Programmer	Date		Key										

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00100	A	*		DISPLAY					F		FILE EXAMPLE								
00101	A	*																	
00102	A																		REF(PAYROLL)
00103	A								R		MENU								
00104	A		N01																
00105	A	O	02								FLDA			20I	20		2	2	DSPATR(HI)
00106	A										FLDB			22N	2B		3	2	
00107	A			72	73														
00108	A	O		60	61			62											
00109	A	A		63															DSPATR(HI)
00110	A										FLDC			7Y	0B		7	20	DSPATR(RI PC)
00111	A			42	43														
00112	A	O		60	61														
00113	A	O		62													9	2	'Constant'
00114	A										FLDD	R					11	2	
	A																		
	A																		

In this coding example, the option indicators cause the following conditions:

- FLDA is displayed and has high intensity if indicator 01 is off or indicator 02 is on.
- FLDB is always displayed. It is displayed with high intensity under either of the following conditions:
  - Indicators 72 and 73 are both on.
  - Indicators 60, 61, 62, and 63 are all on.
- FLDC is always displayed with its image reversed and the cursor positioned to the first position of the field.
- The constant field *Constant* is displayed under any of the following conditions:
  - Indicators 42 and 43 are both on.
  - Indicators 60 and 61 are both on.
  - Indicator 62 is on.
- FLDD is always displayed.

Figure 4-1. Display File Coding Example

Positional Entries (Positions 1-44)

This section describes how to specify the first 44 positions of the Data Description Specifications form for display files. To code the rest of the form, see the section *Keyword Entries (Functions Field, Positions 45-80)* in this chapter.

Figure 4-1 shows some positional entries for display files.

Positions 1-5 (Sequence Number)

Use these positions to specify a sequence number for each line on the form. The sequence number is optional and is used for documentation purposes only.

Position 6 (Form Type)

The A in this position designates this as a Data Description Specifications form. The form type is optional and is for documentation purposes only.

Position 7 (Comment)

An asterisk (*) in position 7 identifies the line as a comment line. Comment lines can appear anywhere in DDS and are retained only in the source file and printed on the source listing. (Comments are not printed on the expanded source listing.) Use positions 8 through 80 for comment text. A blank line (no characters specified in positions 7 through 80) is treated as a comment line.

Positions 7-16 (Conditioning)

Option Indicators

Positions 7 through 16 are a multiple field area in which you can specify option indicators. Option indicators are 2-digit numbers from 01 to 99. Your program can set option indicators on (hex F1) or off (hex F0) to select a field or keyword.

A condition is an ANDed grouping of two through nine indicators that must all be in effect (set off if N is specified; set on if N is not specified) before the field or keyword is selected. You can specify a maximum of nine indicators for each condition and nine conditions for each field or keyword; therefore, a maximum of 81 indicators can be specified for each field or keyword, when nine indicators are used with nine conditions.

When you specify a condition so that more than one indicator must be on or off before the condition is satisfied, it is called an *AND* condition; the first indicator you specified, AND the second, AND the third, and so on, must all be in effect before the condition is satisfied and the field or the keyword is selected. The field or the keyword must be specified on the same line as the last (or only) set of indicators specified.

You can also specify several conditions for a field or keyword so that if any one of them is satisfied, the field or the keyword is selected. This is called an *OR* relationship, and it means that if the first condition is satisfied, OR the second condition, OR the third condition, and so on, the field or the keyword is to be selected. Note that conditions within the OR relationship can consist of just one indicator or can consist of several indicators ANDed together. Indicators can be ANDed to form a condition; conditions can be ORed to give your program several ways to select the field or keyword.

AND (Position 7): If you need more than three indicators to form an ANDed condition, specify them on the next line or lines. You can specify an A in position 7 on the second or following lines to continue the ANDed condition, or you can leave it blank because A is the default.

OR (Position 7): If you specify several conditions that are to be ORed together, each condition must start on a new line and each condition except the first must have an O in position 7. An O specified for the first condition produces a warning message, and that position is assumed to be blank.

NOT (Positions 8, 11, 14): If you want an indicator to be off instead of on to satisfy a condition, specify an N in the position just preceding the indicator (position 8, 11, or 14).

Conditioning More than One Field or Keyword: If you are conditioning a field, the field name (or the constant) and the last (or only) indicator must be on the same line. If the field is not selected for an output operation, no keywords specified for that field are in effect, regardless of how the keywords are conditioned. For example, in Figure 4-1, FLDA is selected if either indicator 01 is off or indicator 02 is on. If FLDA is not selected, any keyword associated with that field, such as DSPATR(HI), is ignored.

If you want to condition one or more keywords, the last (or only) indicator must appear on the same line as the keywords. If the conditioning applies to keywords on more than one line, keyword continuation must be used for the indicators to apply to all keywords. See *Data Description Specifications Syntax Rules* in Chapter 1, *Introduction.*

Screen Size Condition Names

If you want your program to open this file to display devices with screen sizes other than 24 lines by 80 characters, you must specify the DSPSIZ keyword at the file level. You can then condition the use of keywords and the location of fields with the *screen size condition names* specified for the DSPSIZ keyword. If you do not specify the DSPSIZ keyword, your program can only open this file to display devices with a 24 by 80 screen.

The following table shows the screen size condition name for each display device:

Device	**Screen Size**	**Screen Size Condition Name**
System Console	16 lines by 64 characters (1024 characters)	*DS1
5251 (Models 1 and 2) 5252	12 lines by 80 characters (960 characters)	*DS2
5251 (Models 11 and 12)	24 lines by 80 characters (1920 characters)	*DS3
Note: You can specify a user-defined screen size condition name instead of *DS1, *DS2, or *DS3. For an explanation of how to specify user-defined condition names, see the DSPSIZ keyword description.		

The following example shows how to specify the DSPSIZ keyword and screen size condition names:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: & Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blR/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (bl A/P/S/B A/S/X/Y/N/1/W)	Decimal Positions	Usage (bl/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A																		DSPSIZ(24 80 *LARGE 16 64 *CON)
00020	A								R		RECORDA								
00030	A										FIELDA		10	0			1	2	
00040	A										FIELDB		10	0			1	65	
00050	A			*CON													1	49	
00060	A										FIELDC						17	1	
00070	A			*CON													15	1	
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

In this example, the screen size condition name for the primary screen size is defined as *LARGE 1, and the screen size condition name for the secondary screen size is defined as *CON 2. FIELDA appears on line 1, position 2 for both screen sizes. FIELDB appears on line 1, position 65 for the primary screen size (*LARGE by default), and on line 1, position 49 for the secondary screen size (*CON specified in positions 9 through 16). FIELDC appears on line 17, position 1 for the primary screen size and on line 15, position 1 for the secondary screen size. Note that only secondary screen sizes (in this example *CON) can be used to condition field locations.

You use these screen size condition names in a way similar to that of option indicators, except that screen size condition names do not appear in your program and do not appear in the output record. A screen size condition name is on if the display file is opened to the corresponding display size. When you use screen size condition names, the following rules apply:

- Specify the DSPSIZ keyword to designate the primary screen size and any secondary screen sizes. If the DSPSIZ keyword is not specified, the default is DSPSIZ(*DS3).

- You can specify only one screen size condition name for a condition. (AND or an OR with other screen size condition names or option indicators is not permitted.)

- The screen size condition name must start in position 9.

- The screen size condition name can be user-defined. See the DSPSIZ keyword description for more details.

- You can specify N in position 8 to designate a NOT condition (for the primary screen size). Note that this *implies* an OR relationship between the remaining screen size condition names (for example, N*DS1 implies *DS2 or *DS3 when *DS2 and *DS3 are specified as secondary screen sizes on the DSPSIZ keyword).

- You must not use screen size condition names so that the line or position sequence of a field within a record is altered. Fields are ordered in the display file by primary locations; should any secondary location alter this primary sequence, a severe error occurs at file creation time.

  For example, FLD1 and FLD2 are on the primary screen located on line 2, position 2 and line 4, position 2, respectively. It is invalid to use a screen size condition name to display FLD2 ahead of FLD1 on the screen (on line 1) for a secondary screen size.

- When you specify the location of a field on a secondary screen size, you can only specify positions 8 through 16 (conditioning) and 39 through 44 (location).

- If you do not specify a condition name for a keyword for which condition names are valid, the primary condition name specified on the DSPSIZ keyword is the default.

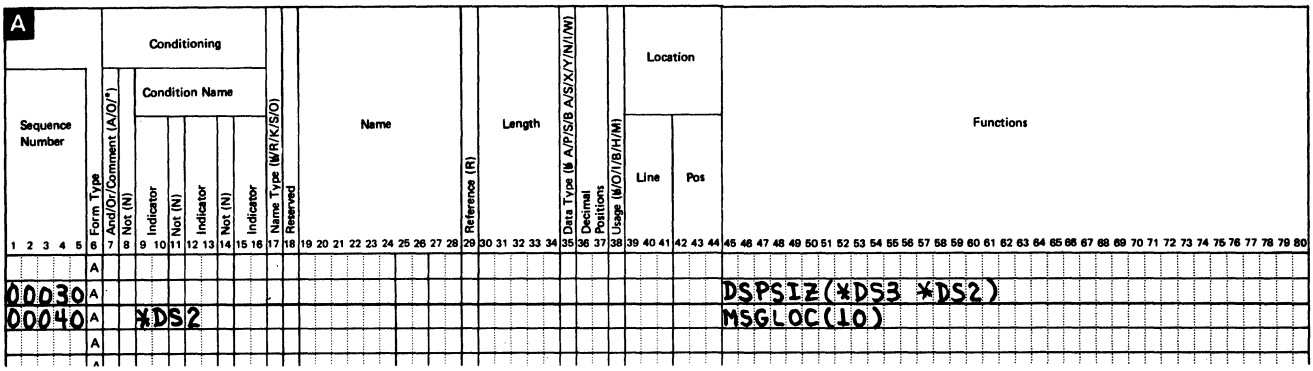
Figure 4-2 shows the valid and invalid combinations of screen size condition names and primary screen sizes, when all screen sizes have been specified on the DSPSIZ keyword and the first one specified varies.

Screen Size Condition Name[1]	Primary Screen Size		
	16 by 64 DSPSIZ(*DS1...) or DSPSIZ(16 64...)	**12 by 80 DSPSIZ(*DS2...) or DSPSIZ(12 80...)**	**24 by 80 DSPSIZ(*DS3...) or DSPSIZ(24 80...)**
*DS1	ERROR[2]	VALID	VALID
*DS2	VALID	ERROR[2]	VALID
*DS3	VALID	VALID	ERROR[2]
N*DS1	VALID	ERROR[3]	ERROR[3]
N*DS2	ERROR[3]	VALID	ERROR[3]
N*DS3	ERROR[3]	ERROR[3]	VALID

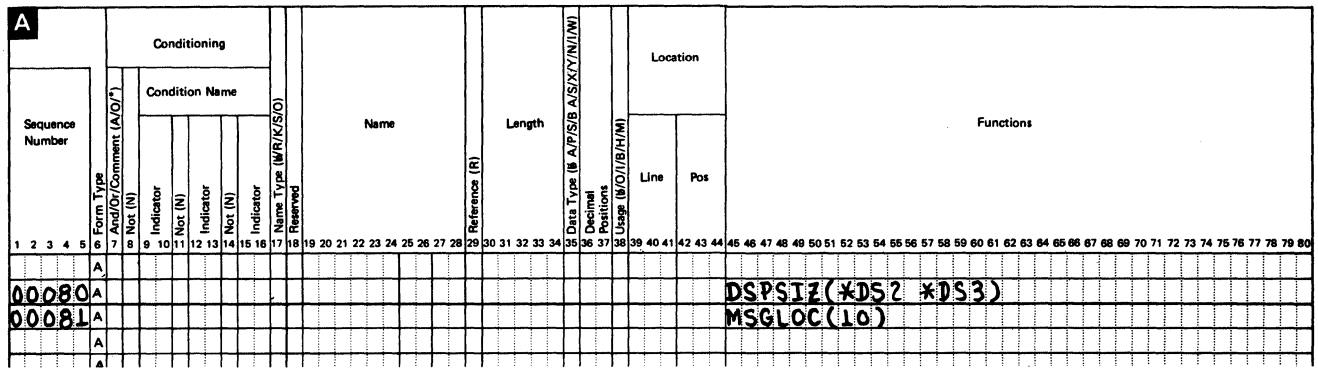
[1]See the DSPSIZ keyword description for providing user-defined names for these screen size condition names.
[2]These screen size condition names are in error because that screen size is the primary screen size.
[3]These screen size condition names are in error because a primary and a secondary location are implied for the same screen size. A condition name specified with the NOT condition implies an OR relationship. For example, N*DS1 implies *DS2 *DS3.

Figure 4-2. Valid Screen Size Condition Specifications

The following two examples show screen size conditioning for a keyword:



Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00030	A																		DSPSIZ(*DS3 *DS2)
00040	A			*DS2															MSGLOC(10)



Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00080	A																		DSPSIZ(*DS2 *DS3)
00081	A																		MSGLOC(10)

In the first example, the screen size condition name *DS2 is specified, so that the message line is line 10 for a 12 by 80 screen and line 24 (the default) for a 24 by 80 screen. In the second example, the message line is also line 10 for the 12 by 80 screen and line 24 for the 24 by 80 screen, even though no screen size condition name was specified, because the primary screen size (*DS2) specified with the DSPSIZ keyword is the default.

Position 17 (Name Type)

The value in this position identifies the type of name specified in positions 19 through 28 (name). The valid entries for display files are:

Entry	Meaning
R	Record format name
Blank	Field name

Position 18 (Reserved)

This position does not apply to any file and must be blank unless an asterisk appears in position 7 (indicates a comment line).

Positions 19-28 (Name)

Use these positions to specify record format names and field names. The following rules apply:

- Names must be 10 characters or less.
- Names must start in position 19.
- Names must begin with an alphabetic character (A through Z, @, $, and #). All subsequent characters can be alphameric (A through Z, 0 through 9, @, $, #, and underscore (_)). There can be no embedded blanks.

Figure 4-1 shows how to specify record format names and field names.

High-level languages can impose specific length and value restrictions on names. For example, RPG III accepts only field names of 6 characters or less and record format names of 8 characters or less, and they cannot contain an underscore. It is your responsibility to ensure that the names used here are acceptable to all language processors that process the file.

Record Format Name: When you specify R in position 17 (name type), the name specified in positions 19 through 28 is a record format name. You can specify more than one record format for a display file, but each record format name must be unique within that file.

Field Name: When position 17 (name type) is left blank, the name specified in positions 19 through 28 is a field name. Field names must be unique within the record format.

Constant Fields: Constant fields are unnamed fields (positions 19 through 28 must be blank). Further, the following rules apply to constant fields:

- Position 17 through 38 must be blank.
- The location of the field is required (positions 39 through 44).
- The field can be conditioned using option indicators (positions 7 through 16).
- Secondary screen locations can be specified using screen size condition names (positions 8 through 16). Only the screen size condition name and location can be specified; that is, positions 7, 17 through 38, and 45 through 80 must be blank.
- The constant itself is defined in positions 45 through 80 using one of the following entries:
  - Explicit DFT keyword (specify the literal within apostrophes with the DFT keyword)
  - Implicit DFT keyword (specify the literal within apostrophes without the DFT keyword)
  - DATE keyword (specify no literal; see the DATE keyword description)
  - TIME keyword (specify no literal; see the TIME keyword description)
- The EDTCDE or EDTWRD keyword can be specified only when DATE or TIME is also specified.

How to Determine the Order of Fields in a Record Format: The order in which you specify named fields in a record format is the order in which they appear in your program when it is compiled. (Unnamed fields do not appear in your program.)

The order in which named and unnamed fields appear in the actual display is determined by the locations you specify in positions 39 through 44. See *Positions 39-44 (Location)* later in this chapter. Hidden fields (H in position 38) do not appear on the display.

Position 29 (Reference)

Specify R in this position to use the reference function of CPF to copy the attributes of a previously defined named field (called the *referenced field*) to the field you are defining. The referenced field can be previously defined in the display file you are defining; or it can be in a previously created data base file (the data base file to be referenced is specified with the REF or REFFLD keyword). The attributes referenced are the length, data type, and decimal positions of the field, as well as editing, validity checking, and text keywords.

If you do not specify R, you cannot use the reference function for this field and you must specify field attributes for this field.

Position 29 must be blank at the file and record levels.

The name of the referenced field can be either the same as the field you are defining or different from the field you are defining. If it is the same, you need only specify R in position 29 (in addition to specifying the name of the field you are defining in positions 19 through 28). If it is different, you must specify the name of the referenced field with the REFFLD keyword.

You can specify the name of the file defining the referenced field as a parameter value with the REF or the REFFLD keyword. See REF and REFFLD keyword descriptions later in this chapter and Appendix A, *How to Specify REF and REFFLD*, for explanations of how CPF finds the referenced field.

You do not need to copy all attributes from the previously defined field to the field you are defining. To override specific attributes of the referenced field, specify those attributes for the field you are defining as follows:

- To override editing keywords (EDTCDE or EDTWRD), specify EDTCDE or EDTWRD for the field you are defining. (You can simply delete these keywords by specifying DLTEDT for the field you are defining.)

- To override validity checking keywords (CHECK, COMP, RANGE, VALUES), specify any validity checking keyword for the field you are defining. (You can simply delete these keywords by specifying DLTCHK for the field you are defining.)

When you override some specifications, others are also affected, as follows:

- If you specify keyboard shift attribute, field length, or decimal positions for the field you are defining, neither editing nor validity checking keywords are copied from the referenced field.

- If you override the previously defined data type to character (by specifying A, X, or W in position 35), decimal positions are not copied; however, if you specify N or I in position 35 and leave blanks in positions 36 through 37 (decimal positions), the field you are defining has the decimal positions of the referenced field.

- Packed decimal and binary fields are not supported for display files; therefore, when you reference fields of these types, the data type assigned is zoned decimal with a keyboard shift as follows:
  - If editing is in effect for the field you are defining, the keyboard shift is numeric only (Y in position 35).
  - If no editing is in effect for the field you are defining, the keyboard shift is signed numeric only (S in position 35).

Note: Once the display file is created, the referenced file can be deleted or changed without affecting the field definitions in the display file. To incorporate changes made in the referenced file, delete and re-create the display file.

Positions 30-34 (Length)

You must specify a length for each named field unless the length is being copied from a referenced field. The length is the number of bytes of data to be passed to or received from your program when I/O operations are done for the field. This is called the *program length* of the field.

The length of a field when it appears on the display is called the *display length*; it is greater than or equal to the program length. The display length of a field is determined by the keyboard shift (specified in position 35) and other field specifications such as decimal positions (positions 36 through 37) and editing functions. (See *Position 35 (Data Type/Keyboard Shift)* for more information.)

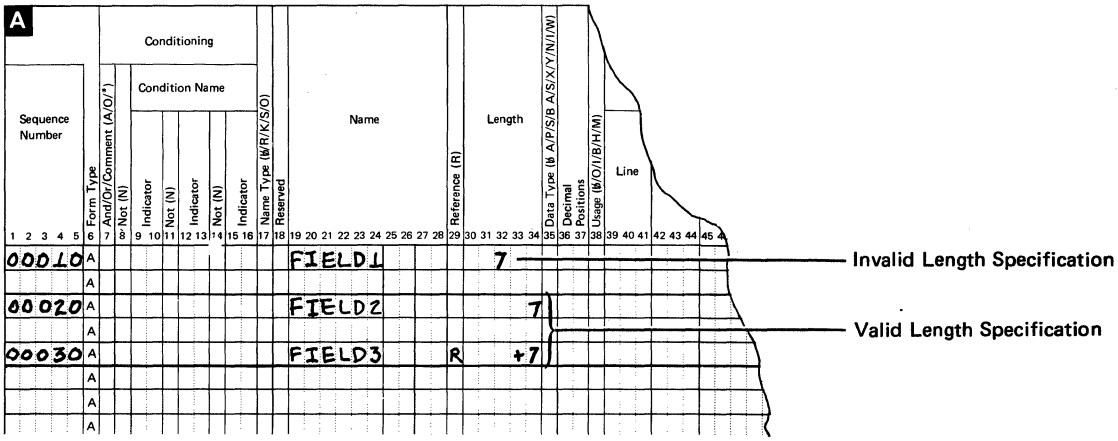
The display length does not include beginning and ending attribute characters of a field; however, these characters must be considered when planning the display layout for field locations. Within a record, the ending attribute character of a field can overlap the beginning attribute character of the next field, requiring only one space between fields. See *Positions 39 through 44 (Location)* for more information.

The maximum length of a character field is equal to the display size minus two. (This allows space for the beginning and ending attribute characters.) The maximum length of a numeric (zoned decimal) field is 31 positions.

A field length must not be specified for a constant field. See the DATE, DFT, or TIME keyword descriptions for explanations of the lengths of constant fields.

When you specify the length of a field, the entry must be right-justified; leading zeros are optional.

The following example shows valid and invalid field length specifications:



If you are using a referenced field, you can override the length of the field by specifying a new value or by specifying the increase or decrease in length. To increase the length, specify +n, where n is the increase. To decrease the length, specify -n, where n is the decrease. For example, an entry of +4 for a numeric field indicates that it is to be 4 digits longer than the referenced field.

A field cannot occupy the first or the last position on the display. These positions are reserved for attribute characters. For example, on a 24 by 80 screen, an entry of 5 in positions 30 through 34 (length), 24 in positions 39 through 41 (line), and 75 in positions 42 through 44 (position) is invalid for a signed numeric field. The display length is 6, causing the field to end in position 80 (the sign position). This does not allow room for the ending attribute character.

Position 35 (Data Type/Keyboard Shift)

For display files, the entry you make in position 35 is the keyboard shift attribute. This entry does not determine the data type of the field used in your program. The entry in positions 36 through 37 (decimal positions) determines the internal data type of the field.

The keyboard shift attribute automatically shifts 5250 work stations with data-entry keyboards and, for all keyboards, can limit what the work station user can key into a field. However, it does not shift 5250 work stations with the typewriter-like keyboard or the system console. Also, it does not restrict in any way what your program can write to a field. Your program can write alphabetic characters to a numeric field and, in most cases, read that field and receive those characters. Any restrictions are enforced solely by the programming language used for your program.

The following are valid entries for display files:

Entry	Meaning	Data Type Permitted
Blank	Default	
X	Alphabetic only	Character
A	Alphameric shift	Character
N	Numeric shift	Character or numeric
S	Signed numeric	Numeric
Y	Numeric only	Numeric
W	Katakana (for Japan only)	Character
I	Inhibit keyboard entry	Character or numeric

Figures 4-1 and 4-3 show how to specify the keyboard shift attribute.

The following sections describe the keyboard shift attributes in detail.

Default (blank): If you leave position 35 blank, the entry in positions 36 through 37 (decimal positions) determines the internal data type of the field as follows:

- If you make a valid entry in positions 36 through 37, the data type is zoned decimal and the keyboard shift attribute is signed numeric (S). This is not true if you also specify an editing keyword; in that case, the keyboard shift attribute is numeric only (Y).
- If you make no entry in positions 36 through 37, the data type is character and the keyboard shift attribute is alphameric shift (A).

If a referenced field is defined as packed or binary, it is converted to zoned decimal in the display file. Conversion to or from packed or binary can occur within your program.

Alphabetic Only (X): Both types of keyboards are in lowershift. Only the characters A through Z, comma (,), period (.), dash (–), and space ( ) can be keyed in. When lowercase characters a through z are keyed in, uppercase characters are sent to the program. (See the CHECK(LC) keyword description for an explanation of how to permit keying in of lowercase characters for the typewriter-like keyboard on the 5250 work station.)

Alphameric Shift (A): Both types of keyboards are in lowershift. All characters are valid for entry.

Numeric Shift (N): Each type of keyboard is shifted to allow numeric entry: uppershift for the data-entry keyboard and lowershift for the typewriter-like keyboard. All characters are valid for entry.

The display length for a numeric shift field is one more than the length coded in positions 30 through 34 when the following conditions occur:

- The field is an unedited, input-capable field.
- The value in the decimal positions field is greater than zero.

The extra position in the display length is for the decimal point.

Note: Input numeric shift fields that have decimal positions specified (in positions 36 and 37) are processed by data management as numeric-only fields, except that editing is not supported. For more information, see the keyboard shift attribute *Numeric Only (Y)*, described later in this section.

Signed Numeric (S): The data-entry keyboard is in lowershift; the typewriter-like keyboard is in uppershift. Only the numbers 0 through 9 can be keyed into the field (no blanks, no plus sign, no minus sign). To leave the field, press the Field Exit key, the Field + key, the Field - key, a cursor movement key, or the Enter/Rec Adv key.

When choosing between signed numeric and numeric only, you should consider these differences:

- Signed numeric restricts the characters that can be keyed into the field to the numbers 0 through 9.
- Signed numeric does not permit EDTCDE or EDTWRD keywords.
- Numeric only performs character removal to remove nonnumeric characters; signed numeric prevents the work station user from keying in these characters at all.

The display length for the field is one more than the length specified in positions 30 through 34. The rightmost position on the screen is reserved for a minus sign.

When CPF passes the contents of the field to your program, the following considerations apply:

- Your program always sees a numeric, right-adjusted, zero-filled field.
- The field is displayed as a right-adjusted, blank-filled field (unless CHECK(RZ) is specified, in which case it is displayed as right-adjusted, zero-filled).
- CPF does not perform decimal alignment.
- CPF does not remove characters from the field (as it does for numeric only fields).

When an input-capable signed numeric field is displayed and CHECK(RZ) is not specified, CPF performs zero suppression by default (the EDTCDE and EDTWRD keywords are not valid for signed numeric fields).

Negative numbers are handled as follows:

- On input, the work station user must key in the number and press the Field - key. The number is right-adjusted in the displayed field with a minus sign in the rightmost position. CPF converts the rightmost significant digit to hex Dn, where n is the significant digit, before passing the number to your program. For example, if the work station user keys in 12345 and presses the Field - key, 12345- is displayed and your program sees F1F2F3F4D5.
- On output, CPF converts hex D in the rightmost digit to hex F (thus changing the negative number to a positive number for display purposes) and displays a minus sign in the rightmost (additional) position in the displayed field. For example, if your program sees F1F2F3F4D5, it appears on the display as 12345-.

On the system console, a signed numeric field is treated as a numeric-only field.

For examples of signed numeric fields, and sample data keyed into them, see Figure 4-3 and the table following it later in this section.

Numeric Only (Y): Each keyboard is shifted to allow numeric entry: uppershift for the data-entry keyboard and lowershift for the typewriter-like keyboard. Only the numbers 0 through 9, plus (+), minus (-), period (.), comma (,), and space ( ) can be keyed into the field. To leave the field, any key can be pressed.

The display length for a numeric-only field is one more than the program length (the length specified in positions 30 through 34) when both of the following conditions occur:

- The field is an unedited, input-capable field.
- The value in positions 36 through 37 (decimal positions) is greater than zero.

The extra position in the display length is for the decimal point.

When CPF passes the contents of the field to your program, the following considerations apply:

- Your program sees a numeric, decimally aligned field.
- To enter digits to the right of the decimal, positions 36 through 37 (decimal positions) must be greater than zero and the work station user must key in the decimal character.
- The work station user cannot key in the maximum number of digits, a decimal character, and a sign character, because the display length of the field equals only the program length plus one. (The work station user can avoid keying a sign character by pressing the Field + key or the Field - key.)
- CPF removes all characters except 0 through 9 (whether keyed or supplied through the EDTWRD keyword) and the sign.
- CPF converts embedded blanks (hex 40) to zeros (hex F0) before decimal alignment. (Embedded blanks are blanks between any significant digits in the field; leading blanks, trailing blanks, zeros, plus, and minus are not treated as significant digits. Note that embedded ampersands in an edit word are also converted to zeros before decimal alignment.)
- All nonnumeric characters are removed before decimal alignment and validity checking. (If you specify the EDTWRD keyword and it supplies additional numeric characters (0 through 9), they are not removed.)
- The field length in the input buffer is the program length.

When CPF displays a numeric-only field, the EDTCDE or the EDTWRD keyword, if specified, applies (EDTCDE and EDTWRD can be specified only for numeric-only fields). The display length equals the program length plus the editing characters from the specified edit code or edit word.

Negative numbers are handled as follows:

- On input, the work station user can enter a negative number in one of two ways:
  - Key in the digits, then a minus. The minus sign (-) appears (hex 60) on the display where it was keyed in.
  - Key in the digits, then press the Field - key. If CHECK(RZ) or CHECK(RB) is not specified, a bracket (]) is displayed in the rightmost position. This causes an error message to be displayed at the work station if decimal positions other than zero are specified in positions 36 through 37. If CHECK(RZ) or CHECK(RB) is specified, the keyed in digits are right-adjusted. No minus sign is displayed in either case.

    When a negative number is passed to your program, CPF converts the rightmost significant digit from hex Fn (positive) to hex Dn (negative), where n is the significant digit.

- On output, the sign appears in the rightmost display position, taking up one of the positions in the display length. See EDTCDE and EDTWRD keyword descriptions in this chapter.

Note: CPF examines each character of a numeric-only field to remove nonnumeric characters (plus sign (+), minus sign (-), comma (,), and decimal point(.)) and nonsignificant digits, and to convert embedded blanks to zeros. This can delay response time if a significant number of these fields must be processed on an input operation.

Katakana (W): This field attribute designates the Japanese Katakana keyboard shift. All characters are valid for entry.

Inhibit Keyboard Entry (I): A field with this keyboard shift attribute does not accept data keyed in from the keyboard, and an error is issued if any keys are pressed. The work station user can press the field advance key (or) to position the cursor at the start of the field. This field can be used to allow input from feature devices such as the Magnetic Stripe Reader or the Selector Light Pen. The Field +, Field Exit, and Dup keys are valid for a field with this attribute, and they function the same as if pressed in any input field for which DSPATR(PR) is not in effect.

The display length for an inhibit keyboard entry field is one position greater than the length coded in positions 30 through 34 when the following conditions occur:

- The field is an unedited, input-capable field.

- The value in the decimal positions field is greater than zero.

The extra position in the display length is for the decimal point.

Figure 4-3 and the table following it show how data keyed in by the work station user is passed to your program.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050*
Printed in U.S.A.

File | Programmer | Date | Keying Instruction | Graphic | Key | Description | Page | of

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	
00100	A	X																		
00200	A	X	KEYBOARD SHIFT ATTRIBUTES																	
00300	A										CHARA		5			I	2	2		
00400	A										CHARB		5			I	3	2	CHECK(RB)	Alphameric shift
00500	A										CHARC		5			I	4	2	CHECK(RZ)	
00600	A										CHARD		5	X		I	4	30		Alphabetic only
00700	A										SIGN1		5		0	I	5	2		Signed numeric
00800	A										SIGN2		5		2	B	5	30		
00900	A										NBR1		5	Y	0	I	6	2		
01000	A										NBR2		5	Y	2	I	6	15		Numeric only
01100	A										NBR3		5	Y	2	B	6	30	EDTCDE(L)	
01200	A										NBR4		5	N	2	I	6	40		Numeric shift
01300	A										NBR2		5			H				Hidden field
	A																			
	A																			
	A																			
	A																			
	A																			
	A																			
	A																			

*Number of sheets per pad may vary slightly.

Figure 4-3. Data Type and Keyboard Shift Coding Example

In the following table, three special characters are used:

- The __ means that no character was entered by the work station user.
- The ƀ indicates a blank.
- The] is represented internally as hex D0.

Except where indicated (for instance, the fourth entry under SIGN1, which the user enters by pressing the Field Exit key), the user enters the data only by pressing a command function key. (This table refers to fields defined in Figure 4-3.)

Field Name (Keyboard Shift)	**As Keyed in by the Work Station User**	**As Passed to Your Program**
CHARA	1. _ _ _ _ _	ƀ ƀ ƀ ƀ ƀ (hex 40)
	2. A B C _ _	A B C ƀ ƀ
	3. A _ C _ _	A ƀ C ƀ ƀ
	4. _ _ _ D E	ƀ ƀ ƀ D E
CHARB (Alphameric Shift)	1. _ _ _ _ _	ƀ ƀ ƀ ƀ ƀ
	2. A B C _ _	ƀ ƀ A B C
	3. A _ C _ _	ƀ ƀ A ƀ C
	4. _ _ _ D E	ƀ ƀ ƀ D E
CHARC (Alphameric Shift)	1. _ _ _ _ _	0 0 0 0 0
	2. A B C _ _	0 0 A B C
	3. A _ C _ _	0 0 A ƀ C
	4. _ _ _ D E	0 0 0 D E
CHARD (Alphabetic Only)	1. _ _ _ _ _	ƀ ƀ ƀ ƀ ƀ
	2. A B C _ _	A B C ƀ ƀ
	3. 4 _ _ _ _	error message
	4. A B C $ _	error message

Field Name (Keyboard Shift)	As Keyed in by the Work Station User	As Passed to Your Program
SIGN1 (Signed Numeric)	1. _ _ _ _ _ _	0 0 0 0 0
	2. 1 2 3 _ _ _	0 0 1 2 3
	3. 1 _ 3 _ _ _	0 0 1 0 3
	4. _ _ _ 4 5 _ (Field Exit key)	0 0 0 4 5
	5. _ _ _ 4 5 _ (Field + key)	0 0 0 4 5
	6. _ _ _ 4 5 _ (Field - key)	0 0 0 4 N (hex F0F0F0F4D5)
	7. 1 2 3 4 5 _	1 2 3 4 5
SIGN2 (Signed Numeric)	1. _ _ _ _ _ _	0 0 0 0 0
	2. 1 2 3 4 _ _	0 1 2 3 4
	3. 1 2 _ _ _ _	0 0 0 1 2
	4. 1 2 _ _ _ _ (Field - key)	0 0 0 1 K (hex F0F0F0F1D2)

Field Name (Keyboard Shift)	As Keyed in by the Work Station User	As Passed to Your Program
NBR1 (Numeric Only)	1. _ _ _ _ _	0 0 0 0 0
	2. 0 0 0 0 5	0 0 0 0 5
	3. 0 0 0 5 _	0 0 0 0 5
	4. 0 0 2 _ _	0 0 0 0 2
NBR2 (Numeric Only)	1. _ _ _ _ _ _	0 0 0 0 0
	2. 0 0 0 0 5 _	0 0 5 0 0
	3. 0 0 5 _ _ _	0 0 5 0 0
	4. 5 _ _ _ _ _	0 0 5 0 0
	5. 0 5 0 0 _ _	5 0 0 0 0
	6. 5 b 5 _ _ _	5 0 5 0 0
	7. 5 b b _ _ _	0 0 5 0 0
	8. 5 5 . 1 _ _	0 5 5 1 0
	9. 5 0 0 0 . _	error message (too many digits)
	10. 5 0 . 0 0 0	error message (too many decimal positions)
	11. 5 5 - _ _ _	0 5 5 0]
	12. 5 _ - - _ _	0 0 5 0]
	13. 5 _ + _ - _	0 0 5 0]
NBR3 (Numeric Only)	Input processing is the same as for NBR2.	
NBR4 (Numeric Shift)	1. _ _ _ _ _ _	0 0 0 0 0
	2. 5 - _ _ _ _	0 0 5 0]
	3. _ 5 _ - _ _	0 0 5 0]
	4. 5 _ + _ _ _	0 0 5 0 0
	5. 5 _ A B C _	0 0 5 0 0
	6. 5 _ K K _ _	5 0 2 0]
	7. 5 _ K A K _	5 0 2 0] See Note.
	8. 5 _ K K A _	0 0 5 0 0 See Note.
NBRZ (Hidden Field)	This is a hidden field and does not appear on the screen.	

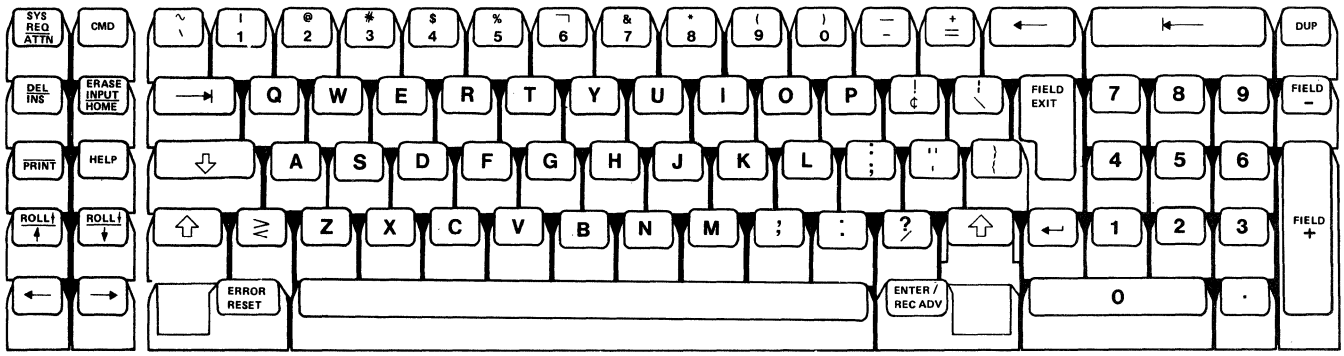
Note: The internal representation of K is hex D2. All nonnumeric characters (including these with hex D in the higher-order position) are deleted with no place value. For example, 5__KAK__ becomes 5020].

Keyboard Types

There are two keyboards on System/38: a *typewriter-like* keyboard and a *data-entry* keyboard. The system console has the typewriter-like keyboard; the 5250 work station can have either the typewriter-like or the data-entry keyboard.

Typewriter-Like Keyboard: The typewriter-like keyboard functions in either uppershift or lowershift. When in uppershift, the upper symbol (for the keys that have two symbols) is entered. When in lowershift, the lower symbol (for the keys that have two symbols) is entered. For alphabetic keys (which have only one symbol), uppercase characters are entered when the keyboard is in uppershift. On the 5250 work station, uppercase alphabetic characters are also entered when the keyboard is in lowershift unless the CHECK(LC) keyword is specified. If the CHECK(LC) keyword is specified, and the work station user places (or leaves) the keyboard in lowershift, lowercase a through z characters can be entered. On the system console, lowercase characters are entered as lowercase; data management translates the entire field to uppercase. Note that none of the keyboard shift attributes causes an automatic uppershift for the typewriter-like keyboard.

The following shows the typewriter-like keyboard:



Data-Entry Keyboard: The data-entry keyboard functions in either numeric shift (upper) or alphabetic shift (lower).

When the keyboard is in uppershift, the upper symbol (for the keys that have two symbols) is entered. On this keyboard, the numbers 0 through 9 are the upper symbols on alphabetic keys. When the keyboard is in lowershift, the lower symbol (for the keys that have two symbols) is entered. The alphabetic characters A through Z are the lower symbols and are always in uppercase. The data-entry keyboard does not support lowercase characters a through z, even if the CHECK(LC) keyword is specified.

The following shows the data-entry keyboard:

Note that you enter the numbers 0 through 9 by using the lowershift on the typewriter-like keyboard and by using the uppershift (numeric) on the data-entry keyboard. Therefore, when a field has one of the numeric keyboard shift attributes (numeric shift or numeric only), it causes the typewriter-like keyboard to be in lowershift and the data-entry keyboard to be in uppershift. In both cases, it allows the entry of numeric characters without having to press a shift key.

Positions 36-37 (Decimal Positions)

Use these positions to specify the decimal placement within a zoned decimal field and also to specify the data type of the field as it appears in your program.

If you leave these positions blank, CPF assigns a data type of character for the field. If you enter a number in these positions, CPF assigns a data type of zoned decimal for the field. The number specified is the number of positions to the right of the decimal point; the entry must be less than or equal to the field length, with a maximum of 31 positions.

If you are using a referenced field, you can override or change these positions. To override decimal positions, specify the new value. To change decimal positions, specify the amount you want the field increased or decreased and precede it with either a plus (+) or minus (-) sign, respectively. For example, an entry of +4 indicates that there are to be four more digits to the right of the decimal point than were in the referenced field.

Figure 4-3 shows how to specify the decimal positions field.

Position 38 (Usage)

Use this field to specify that a named field is to be an output-only, input-only, output/input (both), hidden, or message field. No entry should be made in this position for a constant (unnamed) field. The valid entries for usage and their meanings are:

Entry	Meaning
Blank or O	Output only
I	Input only
B	Output/input (both)
H	Hidden (special output/input field)
M	Message (special output field)

Note: Input-only and output/input fields are *input-capable fields*. Output-only and output/input fields are *output-capable fields*.

Blank or O: If you leave the position blank, output only is the default.

Input Only or Output/Input: All input-capable (I or B) fields are underlined by default. Use the CHGINPDFT keyword or condition the DSPATR (UL) keyword to prevent underlining. See the *CPF Programmer's Guide* for information concerning input-capable field limitations for specific devices.

Hidden: A hidden field is a named, numeric or alphameric field that does not appear on the display. Your program can send data to the field with an output operation, and it can retrieve data from the field with an input operation, but the work station user cannot see or change the contents of the field. The following rules apply to hidden fields:

- Hidden fields are always named.
- Locations are not valid for hidden fields.
- Specify length, data type, and decimal positions as you would for other named fields.
- You can specify more than one hidden field for a display file.

Since hidden fields are not displayed, they are not considered input-capable or output-capable fields, even though your program can send and receive data from them.

Hidden fields are useful in applications involving subfiles. For example, a subfile record can contain record key information in a hidden field. The hidden field cannot be seen by the work station user, but is returned to the program with the subfile record so that the program can return the record to the data base.

Message: A message field is a named, output-only, character field.

The following rules apply to message fields:

- You can choose with option indicators the selection of message fields, but during execution, only one message can be displayed at a time: The message from the first message field selected is displayed, and all others are ignored for that operation.
- When a message field is displayed, all other fields you have specified for that record are processed in the normal manner. If a 5250 work station is being used, the device goes into an error state (locked keyboard, blinking cursor, and message displayed with the high intensity (HI) display attribute). When the work station user presses the Reset key, normal processing continues.
- The text of the message is established when your program moves a value to the message field.
- The location of the message on the display is the message line (the last line on the display unless the MSGLOC keyword is in effect).
- The length you specify for the message field should be less than 64 positions for the system console and less than 79 positions for the 5250 work station. Any message text that occupies more than 63 positions on the system console or more than 78 positions on the 5250 work station is truncated to fit the message line.
- The Help key is not supported for message fields; if it is pressed, an error message is displayed, indicating that no second-level text is available.
- The following keywords are valid for a message field:

  INDTXT
  REFFLD
  TEXT

- You cannot specify M in position 38 for a field if the field is part of the subfile record format.

Note: It is valid to issue an input operation to a record that contains no input-capable fields. This permits pressing a command key as a response to an output record.

Positions 39-44 (Location)

Use these positions to specify the exact location on the display where each field begins (not valid for hidden or message fields). The validity of these entries is based on the DSPSIZ keyword and the screen size condition names. See the DSPSIZ keyword description later in this chapter for specific examples.

Positions 39-41 (Line): Use these positions to specify the line on which the field begins; the entry must be right-justified; leading zeros are optional. The maximum number of lines is 24 or 12 for the 5250 work stations and 16 for the system console.

Positions 42-44 (Position): Use these positions to specify the starting position of the field within the line you specified; it must be right-justified; leading zeros are optional. The maximum rightmost position is 80 for the 5250 work station and 64 for the system console, unless the line specified in positions 39 through 41 is the last line on the screen. When the last line of the screen is specified, the maximum rightmost position is 79 for the 5250 work station and 63 for the system console (the last screen position must be available for an ending attribute character).

For fields other than the first field within the record, you can specify the location by specifying an increment (+n) for positions 42 through 44 (*position*). The increment indicates the number of spaces to be left between the end of the previous field and the start of the field you are defining. An increment of plus zero is not valid.

Beginning Attribute Character: Each field displayed has one attribute character that defines the display attribute of the field on the screen. This attribute character is not displayed, but occupies one position on the screen immediately preceding the field. Because of the beginning attribute character, you cannot specify that a field is to begin in the first position of the screen (line 1, position 1).

When a field begins in position 1 of any line other than line 1, the beginning attribute character occupies the last position of the preceding line. If such a field is the first field of a record, the preceding line is a part of the record area and displays as a blank line. Any record format using that line cannot be displayed at the same time as the other record. The last one to be displayed causes the other one to be erased (unless CLRL(*NO) is specified for the last displayed record).

Ending Attribute Character: The end of a field on the display is indicated by an ending attribute character, unless there is only one space between that field and the next field. In that case, the beginning attribute character of the second field serves as the ending attribute character of the first field. In any case, there must always be at least one space left between fields in a record. When the last position occupied by a field of a record is the last position in a line, the ending attribute character for the field is in position 1 of the next line. However, the next line is not considered part of the first record, and records can be displayed on both lines at once.

Note that the last screen position must be available for an ending attribute character.

Any field that extends beyond the end of a line on the screen is continued on the next line; however, a field cannot extend past the last line on the screen.

Overlapping Fields: Within a record format, you can describe fields to overlap portions of other fields or their attribute characters; however, only one of those fields is displayed on the screen at a time. At execution time, when processing overlapping fields within a record, CPF looks at the fields in row and position sequence. When CPF finds a field whose conditioning is satisfied or that does not have an option indicator specified, it selects that field for display and ignores the rest of the overlapping fields. Note that the first overlapping field that does not have an option indicator specified always stops the search, and any subsequent overlapping fields are never displayed. In the following example, if indicator 01 is set on, field 1 is the only field displayed. If indicator 01 is off and indicator 02 is on, field 2 is the only field displayed. Field 3 is always displayed when neither of the others is selected.

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A			01							FIELD1		10				1	5	TEXT('ONE')
00020	A			02							FIELD2		5				1	5	TEXT('TWO')
00030	A										FIELD3		2		0		1	5	TEXT('THREE')
	A																		
	A																		
	A																		
	A																		
	A																		

If used incorrectly, this capability can result in problems in user and program interface. In the following example, only one input field (FIELD4) is specified for the record, and according to the field location specification, this field overlaps a preceding output field. The work station user is not able to enter any data because FIELD1 is always the field selected for display; the other three fields, including FIELD4, are never displayed.

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A										FIELD1		10				1	5	TEXT('ONE')
00020	A			21							FIELD2		5				1	5	TEXT('TWO')
00030	A			12							FIELD3		2		0		1	5	TEXT('THREE')
00040	A										FIELD4		5Y		21		1	5	TEXT('FOUR')
	A																		
	A																		
	A																		
	A																		

Display Length: The display length is increased for certain types of fields and must be considered when laying out the display. The display length is increased by CPF in the following situations:

- For numeric-only fields with editing, the display length is determined from the edit word or the program length and the edit code.
- For signed numeric fields, the display length is one more than the program length (5250 work stations only).
- The display length is one more than the program length for numeric shift fields and for numeric-only fields without editing when these fields are input-capable and have decimal positions greater than zero.

For an unsigned numeric field (like FIELD4 in the preceding example) with a nonzero decimal position, the system requires a decimal character to be keyed into the field when decimal values are keyed in as data. In the previous example, 123 in FIELD4 does not require a decimal character, but 1234 does (123.4). For this field, the display length is 6.

Keyword Entries (Positions 45-80)

This section contains keyword entries valid for defining display files. They are entered in positions 45 through 80 (functions). See the section *Data Description Specifications Syntax* in Chapter 1 for a discussion of the general rules for specifying keywords. Figure 4-4 shows how to use the general syntax rules for specifying DDS for display files.

IBM International Business Machines Corporation — **DATA DESCRIPTION SPECIFICATIONS** — GX21-7754-1 UM/050* Printed in U.S.A.

File | Programmer | Date | Keying Instruction | Graphic / Key | Description | Page of

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A	*		SYNTAX FOR A DISPLAY FILE															
00020	A	*																	
	A																		
00030	A																		KEYWORDA
	A																		
00040	A								R		RECORDA								KEYWORDB
00050	A																		KEYWORDC KEYWORDD
	A																		
00060	A										FIELDA		20			I	1	3	KEYWORDE('This is a text example')
00070	A																		KEYWORDF(VALUEA)
00080	A																		KEYWORDG(VALUEB VALUEC)
00090	A										FIELDB		40			O	2	3	
00100	A										FIELDC		5		2B		3	3	KEYWORDH('This text example continu-
00110	A																		es with a minus sign')
00120	A																		KEYWORDI('This text example +
00130	A																		continues with a plus sign')
00140	A																4	3	'This literal implies the DFT +
00150	A																		keyword and an unnamed field +
00160	A																		starting at line 4, position 3'
	A																		

You can specify option indicators and screen size condition names in the boxed-in positions.

Figure 4-4. Syntax for a Display File

The following keywords are valid for display files.

ALARM
ASSUME
AUTO
BLANKS
BLINK
BLKFOLD
CAnn
CFnn
CHANGE
CHECK
CHGINPDFT
CLEAR
CLRL
CMP
COMP
CSRLOC
DATE
DFT
DLTCHK
DLTEDT
DSPATR
DSPSIZ
DUP
EDTCDE
EDTWRD
ERASE
ERASEINP
ERRMSG
ERRMSGID
GETRETAIN
HELP
HOME
INDTXT
INZRCD
KEEP
LOCK
LOGINP
LOGOUT
LOWER
MDTOFF
MSGLOC
OPENPRT
OVERLAY
OVRATR
OVRDTA
PASSRCD
PRINT
PROTECT
PUTOVR
PUTRETAIN
RANGE
REF
REFFLD
ROLLDOWN
ROLLUP
RTGAID
RTGCON
RTGDEV
RTGDEVCLS
RTGFIRST
RTGFLD
RTGFMT
RTGPOS
RTNDTA
SETOF
SETOFF
SFL
SFLCLR
SFLCTL
SFLDLT
SFLDROP
SFLDSP
SFLDSPCTL
SFLEND
SFLENTER
SFLINZ
SFLLIN
SFLMSG
SFLMSGID
SFLMSGKEY
SFLMSGRCD
SFLNXTCHG
SFLPAG
SFLPGMQ
SFLRCDNBR
SFLRNA
SFLROLVAL
SFLSIZ
SLNO
TEXT
TIME
UNLOCK
USRDFN
VALUES
VLDCMDKEY

ALARM

Use this record level keyword to specify that CPF is to activate the audible alarm when this record is displayed. For the 5250 work station, the alarm is of short duration. For the system console, the alarm remains on until a record that does not have this keyword specified is displayed.

Option indicators are valid with this keyword.

The following example shows how to specify the ALARM keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (b/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (b A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (b/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00005	A								R		CUST								
00010	A			01															ALARM

ASSUME

Use this record level keyword to specify that CPF is to assume that this record is already displayed on the screen when this file is opened. Such a record would also be defined, with the KEEP keyword, in another display file; that other display file would be closed before this file (in which you are specifying ASSUME) is opened. See the *CPF Programmer's Guide* for information on passing display data between programs.

Specify the ASSUME keyword for at least one record format within the display file so that CPF does not erase the display when the file is opened. In addition, specify the OVERLAY keyword with the ASSUME keyword to prevent CPF from erasing the display when your program issues the first output operation after opening the file.

If you use the ASSUME keyword, at least one field in the record must be able to be displayed. If more than one record with the ASSUME keyword exists, they must occupy unique screen lines.

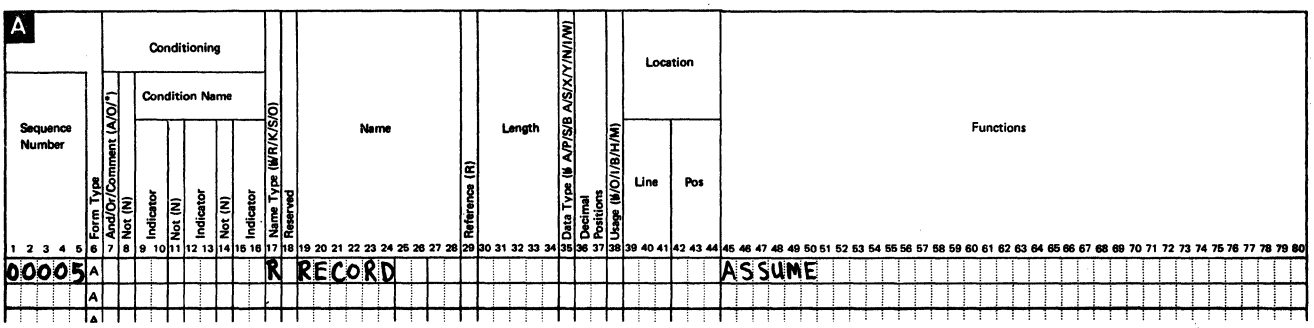
For CPF to process the data correctly, your program must specify the record format name containing this keyword.

The ASSUME keyword is not needed if the record format you are defining is in a shared file (SHARE(*YES) parameter specified on the Create Display File (CRTDSPF), Change Display File (CHGDSPF), or Override with Display File (OVRDSPF) command).

This keyword cannot be specified with the USRDFN or the SFL (subfile record) keyword.

Option indicators are not valid with this keyword.

The following example shows how to specify the ASSUME keyword:



AUTO(RA [RAB | RAZ])
AUTO(RAB | RAZ)

The AUTO keyword is equivalent to the CHECK keyword as follows:

AUTO(RA)	CHECK(ER)
AUTO(RAB)	CHECK(RB)
AUTO(RAZ)	CHECK(RZ)

The CHECK keyword is preferred because it is compatible with DDS on other systems. See *Keyboard Control* in the CHECK keyword description for an explanation of how to use these keywords.

BLANKS(response indicator ['text'])

Use this field level keyword on a numeric, input-capable field to enable your program to distinguish between zero and all blanks on an input operation. When the field contains all blanks, the response indicator is set on. See the example later in this keyword description.

This keyword is also valid for character fields.

The response indicator associated with the BLANKS keyword should be unique within the record. That is, the same response indicator should not be used with other keywords such as CHANGE, DUP, or VLDCMDKEY; with any of the keywords for command keys; or with the BLANKS keyword in other fields in the same record. This is because CPF always turns the response indicator off if the field contains nonblank characters on an input operation. CPF does this to make sure that when the field appears as all blanks, the response indicator is set on, and that when it does not appear as all blanks, the response indicator is set off.

The optional text is included on the listing generated at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The apostrophes are required. If you specify more than 50 characters between the apostrophes, the text is truncated to 50 characters on the program listing.

Option indicators are not valid with this keyword.

The following example shows how to specify the BLANKS keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (b A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (b/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00030	A										QTY1		5	Y	ØB	5	2		BLANKS(Ø1 'ON=QTY1 IS ALL BLANKS')
00040	A										QTY2		5	Y	ØB	6	2		BLANKS(Ø2 'ON=QTY2 IS ALL BLANKS')
00050	A										QTY3		5	Y	ØB	7	2		BLANKS(Ø3 'ON=QTY3 IS ALL BLANKS')
	A																		
	A																		
	A																		

In this example, three numeric fields (QTY1, QTY2, and QTY3) are displayed. If the work station user keys values into the fields and presses the Enter/Rec Adv key, the following occurs:

Value as Keyed into Fields	Value as Passed to Programs	Condition of Response Indicator
100	00100	Off
0	00000	Off
Blanks	00000	On

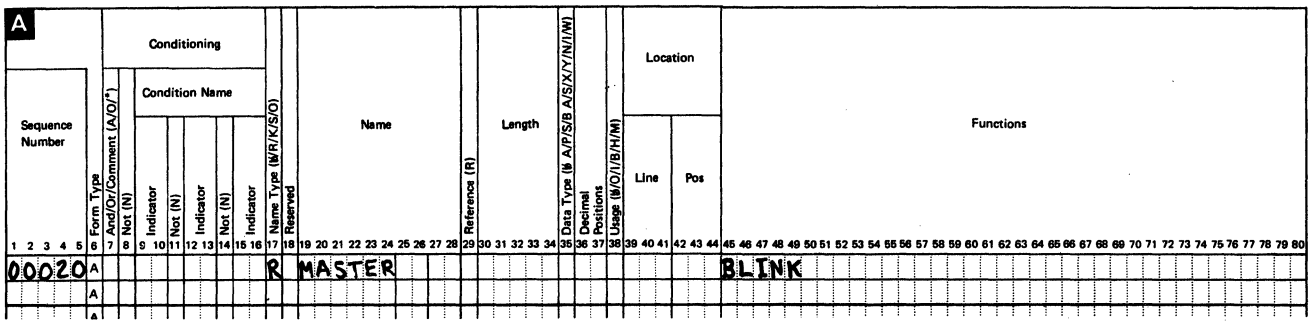
BLINK

Use this record level keyword to specify that as long as the record being defined is displayed, the cursor is to blink. The blinking is set off by the next output operation of a record that does not have the BLINK keyword specified. This keyword is valid for the 5250 work station only; it is ignored for the system console.

The BLINK keyword is ignored at execution time if ALWBLN(*NO) is specified on the Create Device Description (CRTDEVD) or Change Device Description (CHGDEVD) command. See the *CL Reference Manual* for information on how to use these commands.

Option indicators are valid with this keyword.

The following example shows how to specify the BLINK keyword:



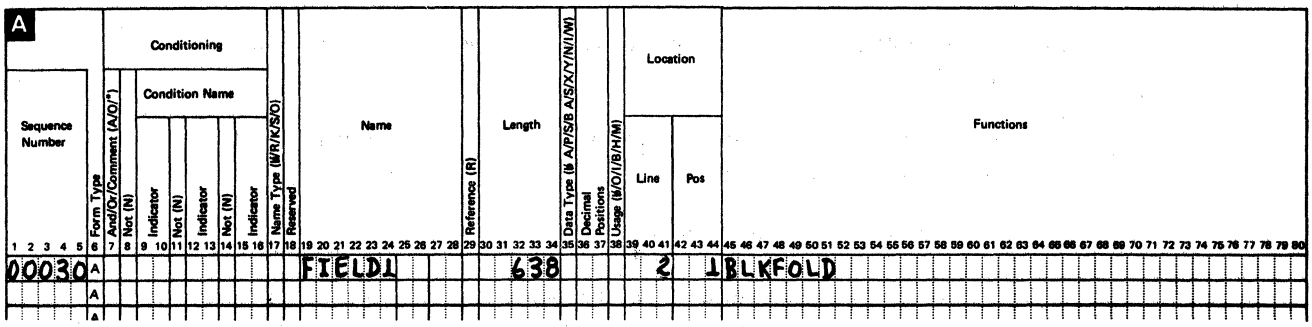
BLKFOLD

Use this field level keyword (blank fold) for named, output-only fields (but not message fields) that are defined so that they overflow onto subsequent display lines. This keyword causes folding to occur at a blank in the data rather than at the end of the display line. It is used to make long text fields easier to read. The default is for the data to be folded at the end of the physical line.

When this keyword is used, the field length is not increased; therefore, it is possible for a portion of the output data to be truncated.

Option indicators are not valid with this keyword.

The following example shows how to specify the BLKFOLD keyword:



CAnn[(response-indicator ['text'])]

Use this file or record level keyword to specify that the command key specified in the keyword (CA01 through CA24) is available for use. It is to be used as a command attention (CA) key; no input data is transmitted from the device. Response indicators 01 through 99 are valid.

If you specify this keyword and the work station user presses the specified command key, the following happens:

- All other command key response indicators in the input buffer are set off (hex F0).
- The response indicator, if specified with the CAnn keyword, is set on (hex F1).
- The CPF data management feedback area (including the routing area) is updated.
- Data already in the input buffer remains unchanged except that the response indicator (if specified) is set on.
- Control is returned to your program.

If you specify a response indicator and the key is pressed, the response indicator is set on and returned to your program. (The text information is associated with the indicator and is used by high-level language compilers to aid in program documentation.)

If the work station user presses a command key and you have not specified it as either a function key (CF) or an attention key (CA), CPF displays a message to the work station user indicating that the key is not valid at that time.

You can use combinations of CA and CF keywords within the same display file, but you cannot specify the same key number as both CA and CF. For example, CA02 and CF02 are not valid in the same display file.

Note: File level CA and CF keys are propagated to the record level. This must be considered when assigning key numbers. For example, if CA02 is specified at file level and CF02 is specified at record level, CF02 is an error.

If you specify a key in the range 1 through 9, you must supply the leading zero in the keyword (for example, CA04).

Option indicators are valid with this keyword.

Validity Checking Considerations: When the work station user presses a CF key, the data from fields with their MDTs set on is placed into the input buffer before validity checking is done. Any errors in the data are then detected, and the appropriate error messages are sent to the display. Because validity checking is not done until after the data is placed in the input buffer, pressing a valid CA key after the CF key can cause invalid data to be returned to your program. This condition is not a problem as long as your program does not process the input data when the CA key is pressed.

This problem can be prevented in either of two ways:

1. Do not allow the use of CA keys. Specify CF keys, which cause validity checking to be done on the data.

2. Do not specify any of the following validity checking keywords if CA keys are allowed:

   CHECK(M10)
   CHECK(M11)
   CHECK(VN)
   COMP (EQ, NE, LT, NL, GT, NG, LE, GE)
   RANGE
   VALUES

Command Keys Valid at Execution Time: As a general rule, the last output operation determines which command keys are valid. However, the following are exceptions to this rule:

- When an operation sends no data to the display screen, the validity of various command keys is not changed. Such operations include:
  - An output operation to a subfile record
  - An update to a subfile record
  - An output operation to a subfile control record that only clears, deletes, or initializes a subfile without displaying the subfile or the subfile control record

- An output operation that displays an error message by selecting ERRMSG or ERRMSGID can also select a CA or CF key to be valid while the error message is displayed.

- If SFLDROP is specified for subfile, the validity of the CA or CF key specified for the SFLDROP keyword is determined by the last output operation. However, as long as the subfile is displayed, the CA or CF key, when valid, acts only as a Drop key.

  If two subfiles with SFLDROP in effect are displayed, only the last SFLDROP keyword to be displayed is in effect and it affects both subfiles.

- If two subfiles with SFLENTER in effect are displayed, only the last SFLENTER keyword to be displayed is in effect. The cursor position at the time the key is pressed determines which subfile is affected.

Notes:

1. The following keywords function like command attention keys:
   CLEAR
   HOME
   PRINT (with response indicator specified)
   HELP
2. CA24 and the Print key are the same on the system console. If both the PRINT and CA24 keywords are specified, pressing CA24 at the system console prints the display and does not perform the functions specified with the CA24 keyword.

The following example shows how to specify the CAnn keyword:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/N)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00011	A																		CA01(91 'End of Program')
00012	A																		CA02(92)
00013	A																		CA03

CFnn[(response-indicator ['text'])]

Use this file or record level keyword to specify that the command key specified in the keyword (CF01 through CF24) is available for use. It is to be used as a function (CF) key to transmit modified data as opposed to an attention key (CA), which does not transmit modified data. Response indicators 01 through 99 are valid.

If you specify this keyword, and the work station user presses the specified command key, the following happens:

- All other command key response indicators in the input buffer are set off (hex F0).
- The response indicator, if specified with the CFnn keyword, is set on (hex F1).
- The CPF data management feedback area (including the routing area) is updated.
- Data is placed in the input buffer according to data received from the device.
- Control is returned to your program.

If you specify a response indicator and the key is pressed, the response indicator is set on and returned to your program along with the input data. If no response indicator is specified, the input data is returned to your program. (The text information is associated with the indicator and is used by high-level language compilers to aid in program documentation.)

If the work station user presses a command key and you have not specified it as either a function key (CF) or an attention key (CA), CPF displays a message to the work station user indicating that the key is not valid at that time.

You can use combinations of CF and CA keywords within the same display file, but you cannot specify the same key number as both CA and CF. For example, CA01 and CF01 are not valid in the same display file.

Note: File level CA and CF keys are propagated to the record level. This must be considered when assigning key numbers. For example, if CA02 is specified at file level and CF02 is specified at record level, CF02 is an error.

If you specify a key in the range 1 through 9, you must supply the leading zero in the keyword (for example, CF03).

Option indicators are valid with this keyword.

Command Keys Valid at Execution Time: As a general rule, the last output operation determines which command keys are valid. The following are exceptions to this rule:

- When an operation sends no data to the screen, the validity of the various command keys is not changed. Such operations include:
  - An output operation to a subfile record
  - An update to a subfile record
  - An output operation to a subfile control record that only clears, deletes, or initializes a subfile without displaying the subfile or the subfile control record

- An output operation that displays an error message by selecting ERRMSG or ERRMSGID can also select a CA or CF key to be valid while the error message is displayed.

- If SFLDROP is specified for a subfile, the validity of the CA or the CF key specified for the SFLDROP keyword is determined by the last output operation. However, as long as the subfile is displayed, the CA or CF key, when valid, acts only as a Drop key.

  If two subfiles with SFLDROP in effect are displayed, only the last SFLDROP keyword to be displayed is in effect and it affects both subfiles.

If two subfiles with SFLENTER in effect are displayed, only the last SFLENTER keyword to be displayed is in effect. The cursor position at the time the key is pressed determines which subfile is affected.

Notes:

1. The following keywords function like command function keys:
   ROLLUP
   ROLLDOWN
2. CF24 and the Print key are the same key on the console. If both the PRINT and CF24 keywords are specified, pressing CF24 at the console prints the display and does not perform the functions specified with the CF24 keyword.

The following example shows how to specify the CFnn keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (b A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (b/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A																		CF01(91 'End of Program')
00020	A																		CF02(92)
00030	A																		CF03

CHANGE(response-indicator ['text'])

Use this record or field level keyword to set on the specified response indicator for an input operation under the following conditions:

- The keyword is specified at the record level, and any input-capable field in the record format has its modified data tag (MDT) set on.
- The keyword is specified for an input-capable field, and that field has its modified data tag (MDT) set on.

The MDT of an input-capable field is set on when the work station user keys into the field or when your program selects the DSPATR(MDT) keyword for the output operation that displays the field. If the MDT is set on using the DSPATR(MDT) keyword, the data in the field may not have changed even though the MDT (and hence the response indicator specified for CHANGE) is set on. Also, note that the MDT is set on even if the work station user rekeys the same data in the field as was initially displayed (such as keying into a blank field and then blanking the field).

Note: The CHANGE response indicator is not set on when a command attention key (CA, Help, Print, Home, or Clear) is pressed.

When CPF detects validity checking errors and redisplays the record with an error message, any CHANGE keyword response indicators that have been set on by keying into fields remain on until all validity checks succeed and the record is passed to your program.

The optional text is included on the listing generated at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The apostrophes are required. If you specify more than 50 characters between the apostrophes, the text is truncated to 50 characters on the program listing.

Option indicators are not valid for this keyword.

The following examples show how to specify the CHANGE keyword:

At the field level:

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (7)	Name Type (17)	Name (19–28)	Length (30–34)	Data Type (35)	Usage (38)	Line (39–41)	Pos (42–44)	Functions (45–80)
00010	A			FLDX	5		B	8	2	CHANGE(67 'FLDX was changed')
	A									
00020	A			FLDY	3		I	8	30	CHANGE(68 'FLDY was entered')
	A									
	A									
	A									

At the record level:

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (7)	Name Type (17)	Name (19–28)	Length (30–34)	Data Type (35)	Usage (38)	Line (39–41)	Pos (42–44)	Functions (45–80)
00010	A		R	REC1						
00020	A									CHANGE(88 'A field was changed')
00030	A	*								
00040	A			FIELD1	10		B	3	2	
00050	A			FIELD2	5		B	5	2	
00060	A			FIELD3	6		B	7	2	
00070	A			FIELD4	3		I	9	2	DFT('ABC')
	A									
	A									
	A									

CHECK(validity-checking-code [. . .])
CHECK(keyboard-control-code [. . .])
CHECK(cursor-control-code)

The CHECK keyword can perform the following functions, depending on the parameter values specified:

Function	Valid Parameter Values
Validity checking	AB, ME, MF, M10, M11, VN
Keyboard control	ER, FE, LC, RB, RZ
Cursor control	RL, RLTB

The following CHECK keywords provide compatibility with other systems:

- CHECK(ER) is equivalent to AUTO(RA)
- CHECK(LC) is equivalent to LOWER
- CHECK(RB) is equivalent to AUTO(RAB)
- CHECK(RZ) is equivalent to AUTO(RAZ)

Option indicators are valid only for CHECK(ER) and CHECK(ME). If specified on a subfile record format, option indicators are valid for these keywords only if the value specified for SFLPAG equals the value specified for SFLSIZ (see the SFLPAG keyword description).

See the following sections for explanations of the CHECK keyword functions.

Validity Checking

Use CHECK at the field level to specify that CPF or the device is to check the validity of the data entered into an input-capable (input-only or output/input) field.

CHECK validates the data by applying one or more edit/check algorithms against the data. An error message is displayed if a specified edit/check algorithm is not satisfied. The valid edit/check codes are:

Edit/ Check Code	Meaning
AB	Allow blanks
ME	Mandatory enter
MF	Mandatory fill
M10	IBM Modulus 10 self-check algorithm
M11	IBM Modulus 11 self-check algorithm
VN	Validate name

(AB) Allow Blanks: Use this code at the file, record, or field level to allow all-blank input to satisfy validity checking for an input-capable field should any associated validity check fail. This enables the passing of data to the program when the work station user has positioned the cursor to the field but left it blank (for instance, by pressing the Erase Input key, the Field Exit key, or the Spacebar). For example, FLD1 is an input-capable field with CHECK(M10 ME) in effect. If the work station user accidentally keys into the field, the M10 algorithm must be satisfied. Specifying (CHECK(M10 ME AB) allows the work station user to blank the field to satisfy validity checking.

When specified at the file level, this keyword is effective for all input-capable fields in the file for which a validity checking keyword is coded. Likewise, when specified at the record level, this keyword is effective for all input-capable fields in the record for which a validity checking keyword is coded. At the field level, this keyword should always be specified with another validity checking keyword (CHECK(M10, M11, VN), COMP, RANGE, VALUES).

CHECK(AB) cannot be used with option indicators and should not be specified if SFLROLVAL or SFLRCDNBR is also specified for the field.

CHECK(AB) can be used in data base files for reference purposes.

When you consider using CHECK(AB) with other validity checking functions, note that processing occurs in the following order:

1. Any of the following:
   a. The keyboard shift attribute specified in position 35 (such as alphameric shift or numeric only) can restrict input keying to certain characters.
   b. If the keyboard shift attribute is numeric shift, the data type (character or numeric) is set by the entry in positions 36 through 37 (decimal positions) and restricts input keying to certain characters.
   c. The CHECK(FE), CHECK(MF), and CHECK(ME) keywords, if specified, restrict input keying.

2. Either of the following:
   a. If CHECK(AB) is specified, data management passes the input data to the program (blanks for a character field and zeros for a numeric field). No further validity checking is done.
   b. If CHECK(AB) is not specified, data management performs the following validity checking functions before passing the data to the program: CHECK(VN), CHECK(M10), CHECK(M11), COMP(. . .), RANGE(. . .), VALUES(. . .).

The following example shows how to specify the CHECK(AB) keyword:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: And (A) or Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/N)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00060	A								R		RECORD1								CHECK(AB)
00070	A										FIELD11		10			B	1	2	TEXT('CHECK(AB) not propagated to t
00080	A																		this field')
00090	A										FIELD21		10			B	1	22	CHECK(VN)
00100	A																		TEXT('CHECK(AB) is propagated to t
00110	A																		this field')
00120	A	*																	
00130	A								R		RECORD2								
00140	A										FIELD12		10			B	2	2	CHECK(VN) CHECK(AB)
00150	A										FIELD22		10			B	2	22	CHECK(VN AB)
00160	A										FIELD32		1			B	2	4	CHECK(AB) VALUES('A' 'B' 'C')
00170	A										FIELD42		10			B	2	62	CHECK(VN)
00180	A																		TEXT('CHECK(AB) is not propagated t
00190	A																		to this field')
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

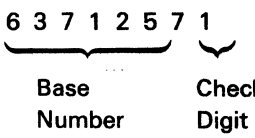
(ME) Mandatory Enter: This code specifies that at least 1 character of data (a blank is valid) must be entered into the field. Note that when no field currently on the display has been modified, the 5250 work station does not enforce mandatory enter. To enforce mandatory enter, specify DSPATR(MDT) for at least one field in each record on the display. For all other fields in the record, CHECK(ME) is then enforced. However, because the device cannot determine if the user has entered data to a field with both DSPATR(MDT) and CHECK(ME), you should also specify DSPATR(ND) so that this field is not displayed.

This code is valid only for the 5250 work station and is ignored for the system console.

(MF) Mandatory Fill: This code specifies that if any part of the field is altered, all positions in the field must have a character entered. Blanks are considered valid characters.

This code is valid only for the 5250 work station and is ignored for the system console. This code cannot be specified with keyboard control codes RB or RZ.

IBM Modulus 10 or 11 Algorithm: This code specifies that data entered into the field must satisfy the IBM Modulus 10 (M10) or Modulus 11 (M11) self-check algorithm. You cannot specify both the Modulus 10 and the Modulus 11 self-check algorithms for the same field. A self-check field is composed of two parts: the base number (for example, your control or account number) and one check digit. The check digit is the rightmost digit in the field. The following is an example of an 8-digit self-check field:



Notes:

1. CPF supports a maximum length of 31 digits for numeric fields.
2. The CHECK(M10) and CHECK(M11) keywords cannot be specified with the COMP(EQ) keyword.

For each position in the base number, there is a Modulus 10 weight factor and a Modulus 11 weight factor. Positions are counted from the rightmost digit (not including the check digit).

The Modulus 10 weight factor is 2 for positions 1,3,5,. . ., 31; it is 1 for positions 2,4,6, . . ., 30. The Modulus 11 weight factors are 2,3,4,5,6,7,2,3,4,5,6,7,. . .,2,3,4,5,6,7,2 for positions 1,2, . . ., 31, respectively.

To compute the Modulus 10 self-check digit, do the following:

1. Multiply the units position and every alternate position of the base number by 2.

2. Add the digits in the products to the digits in the base number that were not multiplied.

3. Subtract the sum from the next higher number ending in zero.

The difference is the self-check digit.

For example:

Base number	6 1 2 4 8
Units position and every alternate position	6 2 8
Multiply by the weight factor, 2	x2 x2 x2
Products	12 4 16
Digits not multiplied	1 4
Add the *digits* of the products and the digits from the base number not used for multiplication	1 + 2 + 4 + 1 + 6 + 1 + 4 = 19
Next higher number ending in 0	20
Subtract	-19
Self-check digit	1

To compute the Modulus 11 self-check digit, do the following:

1. Assign a weight factor to each digit position of the base number. These factors are: 2, 3, 4, 5, 6, 7, 2, 3, 4, 5, 6, 7, 2, 3, , , , starting with the units position of the number and progressing toward the high-order digit. For example, the base number 991246351 would be assigned the weight factors as follows:

   Base number 9 9 1 2 4 6 3 5 1

   Weight factors 4 3 2 7 6 5 4 3 2

2. Multiply each digit by its weight factor.

3. Add the products.

4. Divide this sum by 11.

5. Subtract the remainder from 11.

The difference is the self-check digit.

For example:

Base number	1 3 7 3 9
Weight factors	x6 x5 x4 x3 x2
Multiply each digit by its weight factor	6 15 28 9 18
Add the products	6 + 15 + 28 + 9 + 18 = 76
Divide the sum by 11	76/11 = 6 plus a remainder of 10
Subtract the remainder from 11	11 - 10 = 1
Self-check digit	1

Note: If the remainder in step 4 is 0, the self-check digit is 0. If the remainder is 1, the base number has no self-check digit; you must make sure that such base numbers are not used in the fields you define as self-check fields.

(VN) Validate Name: Use this code to specify that the input field must contain a valid name. The first character must be $, #, @, or A through Z. The remaining characters must be alphameric ($, #, @, A through Z, 0 through 9, or underscore (_)) and must not contain embedded blanks.

The following example shows how to specify the CHECK keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (W A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (W/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00101	A										RESP		2			I	8	2	CHECK(ME MF)
	A																		
	A																		
	A																		
	A																		
	A																		

Keyboard Control

When the CHECK keyword is used with a keyboard control code, it controls certain data-entry aspects at the 5250 work station only. The valid keyboard control codes are:

Keyboard Control Code	Meaning
ER	End of record. Equivalent to AUTO(RA).
FE	Field exit check.
LC	Lowercase. Equivalent to LOWER.
RB	Right-adjust with blank fill. Equivalent to AUTO(RAB).
RZ	Right-adjust with zero fill. Equivalent to AUTO(RAZ).

(ER) End of Record: Use this code to eliminate the need for the work station user to press the Enter/Rec Adv key. Whenever the work station user keys a character (including a blank) into the last position of the field, the record is sent from the device just the same as if the Enter/Rec Adv key had been pressed. If you also specify DSPATR(SP) for the field, the record is sent from the device as soon as the work station user selects the field. If you use this function, it should be on the last field entered by the user for this record.

Option indicators are valid with this keyword.

Note: If specified on a subfile record format, option indicators are valid with the CHECK(ER) function only if subfile page equals subfile size (see the SFLPAG keyword description).

(FE) Field Exit Check: This code specifies that the work station user cannot advance to the next input field without pressing one of the field exit keys. The cursor remains under the low-order character position of the field until a valid field exit key has been pressed, even though that character has been keyed in. If the user presses any other key, an error results.

Valid field exit keys depend on the type of keyboard layout being used. This code applies only to input fields the work station user can key into. This code is valid only for the 5250 work station and is ignored for the system console.

(LC) Lowercase: Use CHECK(LC) for input-only or output/input fields to permit the work station user to enter lowercase a through z. The way the work station user keys in the characters (uppercase or lowercase) is the way the characters appear on the screen and are returned to your program.

Your program can display a field that contains both uppercase and lowercase characters on both the 5250 work station and the system console.

If you specify this keyword, the following occurs when the work station user keys into the field:

- On the 5250 work station, lowercase a through z remain lowercase.
- On the system console, lowercase a through z remain lowercase.

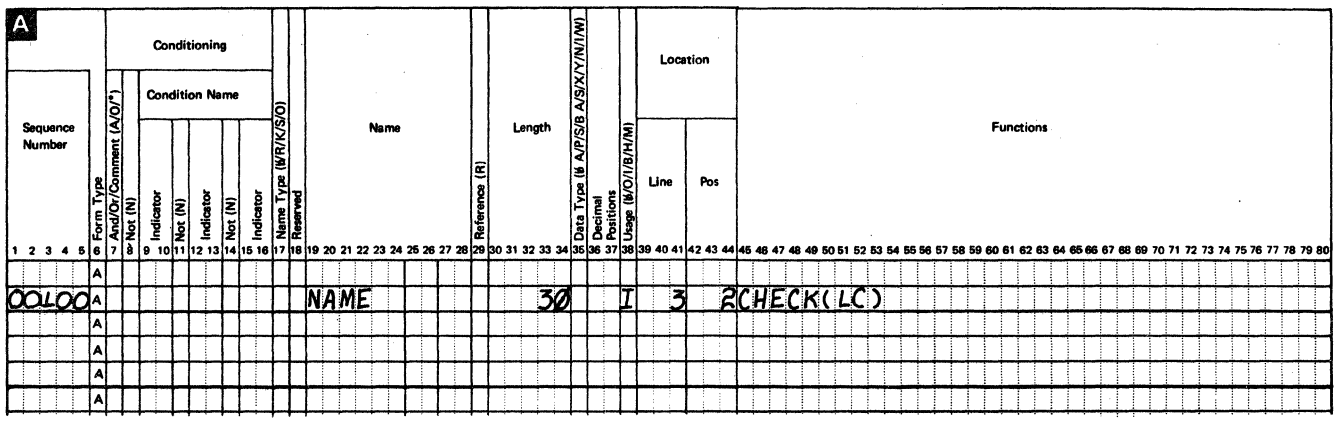
If you do not specify this keyword, the following occurs when the work station user keys into the field:

- On the 5250 work station, lowercase a through z are changed to uppercase.
- On the system console, the entire field is changed to uppercase if any position in the field is keyed into.

The CHECK(LC) keyword has no effect on data-entry keyboards. Data-entry keyboards do not support lowercase characters a through z.

Option indicators are not valid with this keyword.

The following example shows how to specify the CHECK(LC) keyword:



(RB) Right Adjust With Blank Fill: Use this code to specify that data characters entered into this field are to be automatically right-adjusted and the leading characters are to be filled with blanks.

Option indicators are not valid with this keyword.

Notes:

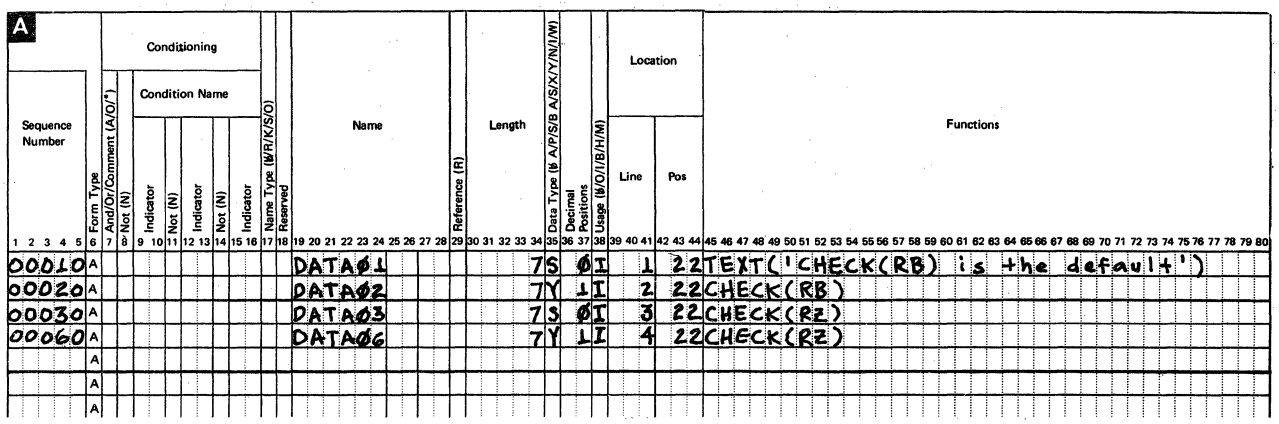
1. You do not need to specify CHECK(RB) for a signed numeric field; when no specification is made, right-adjust with blank fill is the default.
2. CPF automatically performs zero suppression for a signed numeric, input-capable field when CHECK(AB) is specified or is used as the default.
3. A value of zero appears as all blanks on the display.

(RZ) Right Adjust With Zero Fill: Use this code to specify that data characters entered into this field are to be right-adjusted automatically and the leading characters are to be filled with zeros.

A signed numeric field not specified as right-adjust (RB or RZ) is treated as if specified as RB. Right-adjust is activated only by pressing the Field Exit, the Field +, or the Field - key. If a right-adjust field is exited through cursor movement keys, the field is not right-adjusted (it is left as is). Right-adjust fields longer than 15 character positions slow keyboard entries. The Dup key fills a right-adjust field from the cursor location to the end of the field with the duplication character (hex 1C), but the field is not right-adjusted.

Option indicators are not valid with this keyword.

The following example shows how to specify the CHECK keyword for right-adjust with blank fill (RB) and for right-adjust with zero fill (RZ):



Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Pos	Functions
00010	A										DATA01		7	S	0	I	1	22	TEXT('CHECK(RB) is the default')
00020	A										DATA02		7	Y	1	I	2	22	CHECK(RB)
00030	A										DATA03		7	S	0	I	3	22	CHECK(RZ)
00060	A										DATA06		7	Y	1	I	4	22	CHECK(RZ)
	A																		
	A																		
	A																		

This would appear on the screen as follows:

	Data Keyed In	Key Pressed	Result on Screen
DATA01	1 2 3 _ _ _ _ _	Field - key	_ _ _ _ 1 2 3 -
DATA02	_ 1 _ 2 . 3 - _	Field Exit key	_ _ 1 _ 2 . 3 -
DATA03	_ _ 1 2 3 _ _ _	Field + or	0 0 0 0 1 2 3 _
		Field - key	0 0 0 0 1 2 3 -
DATA06	1 . 2 3 4 . 5 _	Field Exit key	0 1 . 2 3 4 . 5

Note: Right-adjust and blank or zero fill depends on the number of positions the work station user keys into and whether or not a screen position is reserved for the sign (signed numeric field).

Cursor Control

When the CHECK keyword is used with a cursor control code, it specifies that the cursor is to move from right to left at a 5250 work station. Right-to-left cursor support requires the right to left feature on your system. See the *CPF Programmer's Guide* for more information concerning this support.

CPF does not ensure that right-to-left files are opened only for work stations capable of right-to-left cursor movement. Therefore, all work stations in the same system should be configured with the same language capability and with the same right-to-left capability.

The valid cursor control codes that can be specified for cursor control are:

Cursor Control Code	Meaning
RL	Right-to-left cursor movement within fields
RLTB	Right-to-left, top-to-bottom cursor movement from field to field

The right-to-left capability includes the following restrictions:

- Right-to-left files cannot be sent to the system console.
- The check digit for modulus checking is the rightmost byte in the field.
- Katakana cannot be used with right-to-left support.
- CHECK(RL) and CHECK(RLTB) cannot be specified with user-defined records (having the USRDFN keyword).

A warning message is issued for the following conditions:

- A right-to-left field that also allows magnetic card reader operator identification data (DSPATR(OID) keyword).
- A right-to-left field that spans more than one line.
- A right-to-left field that is also a self-check field (CHECK(M10) or CHECK(M11) keyword).
- A right-to-left field for which CHECK(RZ) or CHECK(RB) is specified.

Option indicators are not valid with cursor control codes.

(RL) Right to Left: Use the CHECK(RL) keyword at the file, record, or field level to specify that the cursor should move from right to left within input-capable character fields. At the file level, specifying CHECK(RL) makes the cursor move from right to left in all input-capable character fields in the file. At the record level, specifying CHECK(RL) makes the cursor move from right to left in all input-capable character fields in the record. At the field level, specifying CHECK(RL) makes the cursor move from right to left in only the field with which it is associated.

The following example shows how to specify the CHECK(RL) keyword at the file level:

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9–10)	Not (N) (11)	Indicator (12–13)	Not (N) (14)	Indicator (15–16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19–28)	Reference (R) (29)	Length (30–34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36–37)	Usage (b/O/I/B/H/M) (38)	Line (39–41)	Pos (42–44)	Functions (45–80)
	A																		
00010	A																		CHECK(RL)
00020	A								R		DSPLY								
	A																		
	A																		

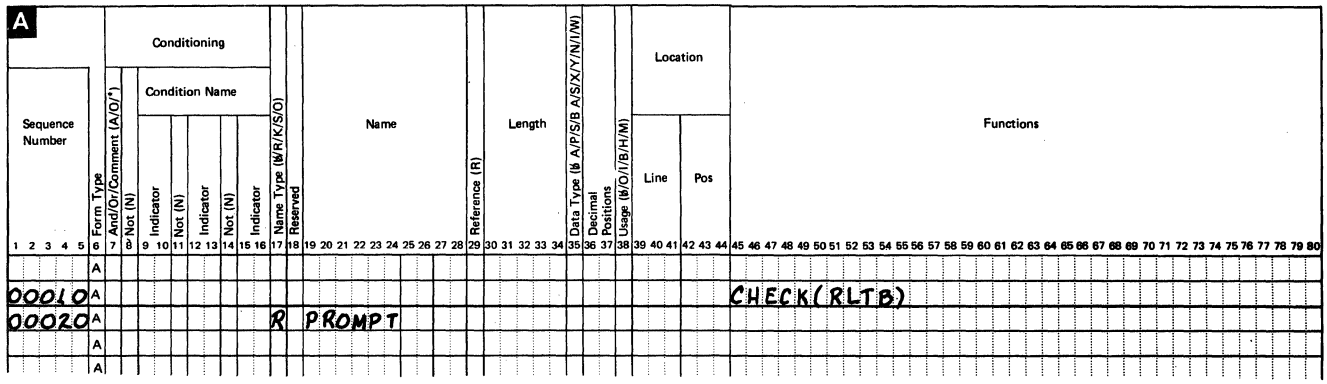
Note: If you want to specify the RL cursor control code with an edit/check code, you can do so only if the edit/check code is valid at the level you specify. In the following example, CHECK(RL AB) is specified at the record level because AB is valid at that level; CHECK(RL MF) is specified at the field level because MF is valid only at that level:

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9–10)	Not (N) (11)	Indicator (12–13)	Not (N) (14)	Indicator (15–16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19–28)	Reference (R) (29)	Length (30–34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36–37)	Usage (b/O/I/B/H/M) (38)	Line (39–41)	Pos (42–44)	Functions (45–80)
	A										{								
	A																		
	A								R		RECORD1								CHECK(RL AB)
	A										{								
	A																		
	A										INPFLD		4			I	4	1Ø	CHECK(RL MF)
	A										{								
	A																		
	A																		
	A																		

(RLTB) Right to Left, Top to Bottom: Use the CHECK(RLTB) keyword only at the file level. It specifies the direction the cursor is to advance from input-capable field to input-capable field. CHECK(RLTB) specifies that upon exiting from a field, the cursor advances by moving from right to left and from top to bottom of the display until it reaches the next input-capable field. You can specify the RLTB cursor control code with only the edit/check code AB, as the others are not valid at the file level.

Note: Specifying CHECK(RLTB) does not change which input-capable field the cursor is positioned in when the display initially appears. See the *CPF Programmer's Guide* for a description of cursor positioning.

The following example shows how to specify the CHECK(RLTB) keyword:



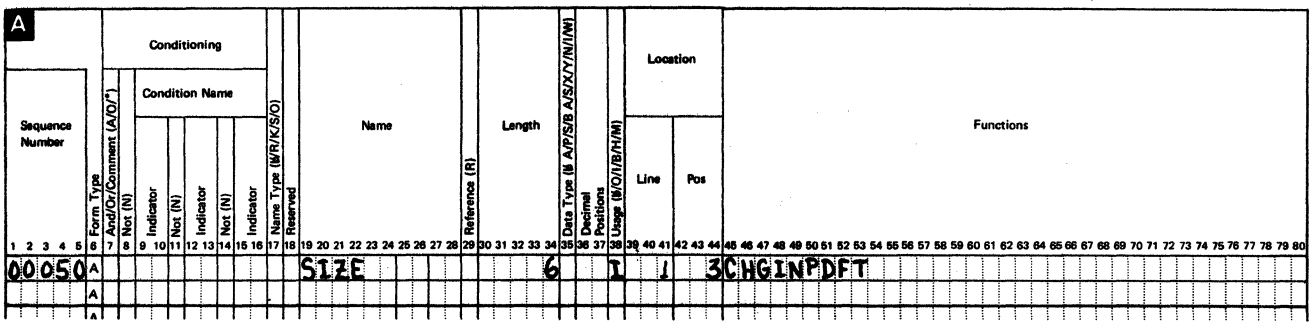
CHGINPDFT

Use this file, record, or field level keyword (change input default) to specify that an input-capable field (input only or output/input) is not to be underlined by default. When used at the file level, this keyword applies to all input-capable fields in a file. When used at the record level, this keyword applies to all input-capable fields in the record.

Option indicators are not valid with this keyword. This keyword does not prohibit you from also specifying the DSPATR(UL) keyword either with or without option indicators.

When DSPATR(UL) is specified, the CHGINPDFT keyword has no effect on the field.

The following example shows how to specify the CHGINPDFT keyword:



CLEAR[(response-indicator ['text'])]

Use this file or record level keyword to specify that your program is to receive control if the work station user presses the Clear key, and optionally, that CPF is to set on the specified response indicator. The Clear key is processed like a CA key (no input data is transmitted from the device). CPF does not clear the screen; your program must perform the desired function (such as clearing fields or records from the screen).

If you do not specify this keyword and the work station user presses the Clear key, CPF displays a message indicating that it is not valid at that time. This keyword is ignored for the system console; it is valid for the 5250 work station only.

Note: On the 5250 work station, the Clear key is pressed by pressing CMD, then the left arrow above the Field Exit key.

The optional text is included on the listing generated at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The apostrophes are required. If you specify more than 50 characters between the apostrophes, the text is truncated to 50 characters on the program listing.

Option indicators are valid with this keyword.

The following is an example of how to specify the CLEAR keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (M A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (M/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00101	A																		CLEAR(10 'Clear key pressed')
	A																		
	A																		
	A																		

CLRL(*NO | nn)

Use this record level keyword (clear line) to specify that before the record is displayed CPF is to clear (erase) a specific number of screen lines. Only those lines are cleared. (Note that, as with OVERLAY, other records remain on the screen.) You can specify one of two parameter values for the CLRL keyword:

- Specify *NO so that no lines on the screen are cleared before displaying the the record whose format you are defining. The displayed record overlays any data already displayed on the screen.

- Specify nn, where nn is an integer between 1 and 24. The number specified is the number of lines cleared, starting with and including the first line on which the record is to be displayed. If the SLNO keyword is also specified for this record format, the clearing of lines begins with the starting line number in effect for the record format at the time it is displayed.

If the record format for which the CLRL keyword is specified has one or more input-capable fields, any records that are overlaid are no longer recognized by CPF. That is, any input-capable fields can no longer be keyed into, any input operation issued to one of those records results in an error, and they cannot be erased by selecting the ERASE keyword. However, if the record format for which the CLRL keyword is specified has no input-capable fields, any records that are overlaid continue to be recognized by CPF. That is, input-capable fields in the overlaid records remain input-capable, and input operations issued to those record formats are still valid. Also, such a record format may not be properly erased when it is overlaid by another record or when it is erased through the use of the ERASE keyword on another record format.

If the CLRL keyword is not specified and neither OVERLAY nor PUTOVR is specified, the entire screen is erased.

If you specify the CLRL keyword, the record format must have at least one displayable field.

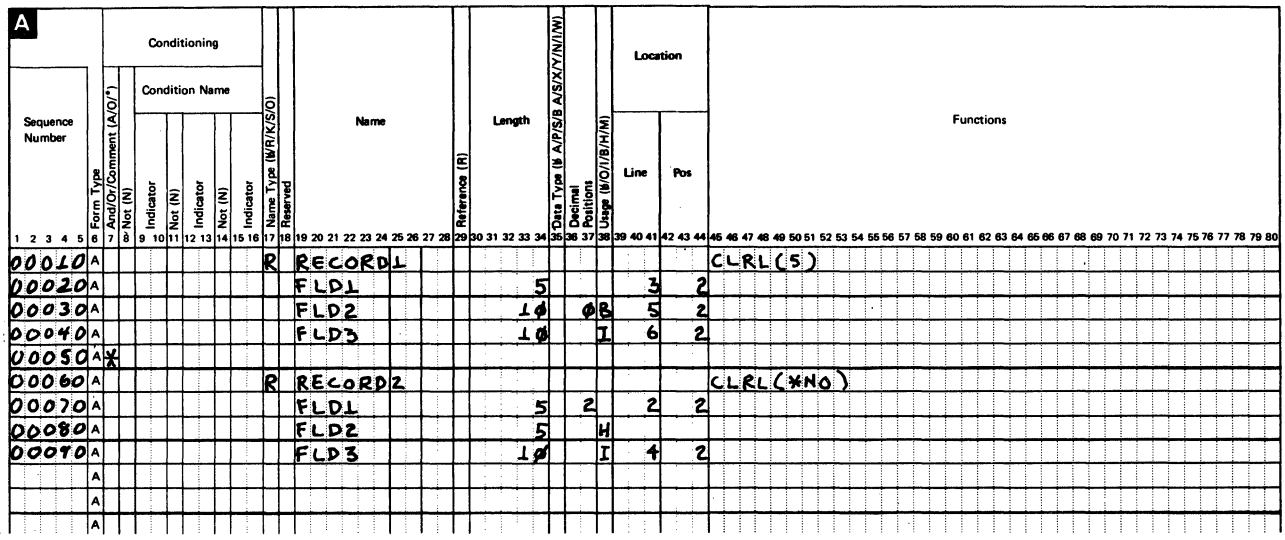
The CLRL keyword cannot be specified with any of the following keywords:

ASSUME
KEEP
SFL
SFLCTL
USRDFN

The CLRL keyword cannot be specified for the record format specified by the PASSRCD keyword.

Option indicators are not valid with the CLRL keyword.

The following example shows how to specify the CLRL keyword:



A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		RECORD1								CLRL(5)
00020	A										FLD1		5				3	2	
00030	A										FLD2		10		0	B	5	2	
00040	A										FLD3		10			I	6	2	
00050	A	*																	
00060	A								R		RECORD2								CLRL(*NO)
00070	A										FLD1		5		2		2	2	
00080	A										FLD2		5			H			
00090	A										FLD3		10			I	4	2	
	A																		
	A																		
	A																		

CMP(relational-operator-code constant-value)

The CMP keyword is equivalent to the COMP keyword. The COMP keyword is preferred because it is compatible with DDS on other systems. See the COMP keyword description for an explanation of how to use these keywords.

COMP(relational-operator-code constant-value)

Use this field level keyword (comparison) to specify that CPF is to compare the data that the work station user enters into an input or output/input field with the specified constant value. The relational operator code is the criterion for the comparison. If the data entered in this field fails this validity check, CPF displays an error message. The valid entries for the relational operator codes are:

Code	Meaning
EQ	Equal
NE	Not equal
LT	Less than
NL	Not less than
GT	Greater than
NG	Not greater than
LE	Less than or equal
GE	Greater than or equal

The specified constant value must be either numeric or character, depending on the data type (decimal positions entry). Numeric constants are expressed by the digits 0 through 9 and a leading sign (+ or -). Character constants must be enclosed in apostrophes.

Note: If the field named (to be compared with) is a numeric field, alignment is based on the decimal positions specified (in positions 36 and 37), and leading and trailing blanks are filled with zeros. If no decimal point is keyed in, the decimal point is assumed to be to the right of the last (rightmost) digit. For example, for a numeric field with a length of 5 (specified in column 34) and 2 decimal positions (specified in column 37), 1.2 is interpreted as 001.20, and 100 is interpreted as 100.00.

You can specify only one operation for the COMP keyword and only one COMP keyword for a field.

Option indicators are not valid with this keyword.

The following example shows how to specify the COMP keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12	13	14 15	16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A										FIELD2		6		0	I	10	10	COMP(EQ +021920)
00020	A										FIELD1		3			I	11	11	COMP(EQ 'ABC')
	A																		
	A																		
	A																		
	A																		

CSRLOC(field-name-1 field-name-2)

Use this record level keyword (cursor location) to specify the cursor location on an output operation to the record format you are defining. Your program issues the output operation after setting the cursor location. The parameter values on the keyword specify the names of two fields whose contents are the line number (for field-name-1) and the position number (for field-name-2) of the cursor location. Field-name-1 and field-name-2 are 3-byte, zoned decimal, hidden fields; your program uses these fields to tell CPF where to locate the cursor.

The cursor is not positioned to the desired location on an output operation that leaves the keyboard locked. The cursor does not move to the desired position until your program issues an input or an output operation that unlocks the keyboard. If your program sets the cursor location fields to values outside the range of values valid for the display device, this keyword is ignored.

For any one output operation, the CSRLOC keyword overrides any other cursor location specifications, such as DSPATR(PC) and SFLRCDNBR(CURSOR), that are in effect. This keyword is in effect until your program issues another output operation with DSPATR(PC), CSRLOC, or SFLRCDNBR(CURSOR) in effect or until the record in which this keyword is specified is overlaid (OVERLAY keyword) or erased (ERASE keyword), whichever comes first.

Specify the CSRLOC keyword only once per record format.

The CSRLOC keyword is not valid for the following record formats;

- Subfile record formats (identified by the SFL keyword)
- User-defined record formats (identified by the USRDFN keyword)

Option indicators are valid with this keyword. Screen size condition names are not valid.

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		RECORD1								CSRLOC(LINNBR POSNBR)
00020	A										TITLE		40			B	1	2	
00030	A										PAGE		5	Y	0	B	1	60	
00040	A										TEXT		1760			B	2	1	
00050	A										LINNBR		3		0	H			
00060	A										POSNBR		3		0	H			
	A																		
	A																		
	A																		

In this example, the application program sets the contents of LINNBR and POSNBR before issuing an output operation to RECORD1. When the record is displayed, the fields TITLE, PAGE, and TEXT appear on the screen. The cursor could be at some location in a field of RECORD1 where the work station user is to make changes.

DATE

Use this field level keyword to specify that CPF is to retrieve the current job date (the date the job was started) and display it as a constant field. The following information is implied for the field, and you cannot specify it:

- The output-only attribute
- A field length of 6 positions

You can specify only the location of the field on the screen. Positions 17 through 38 must be blank.

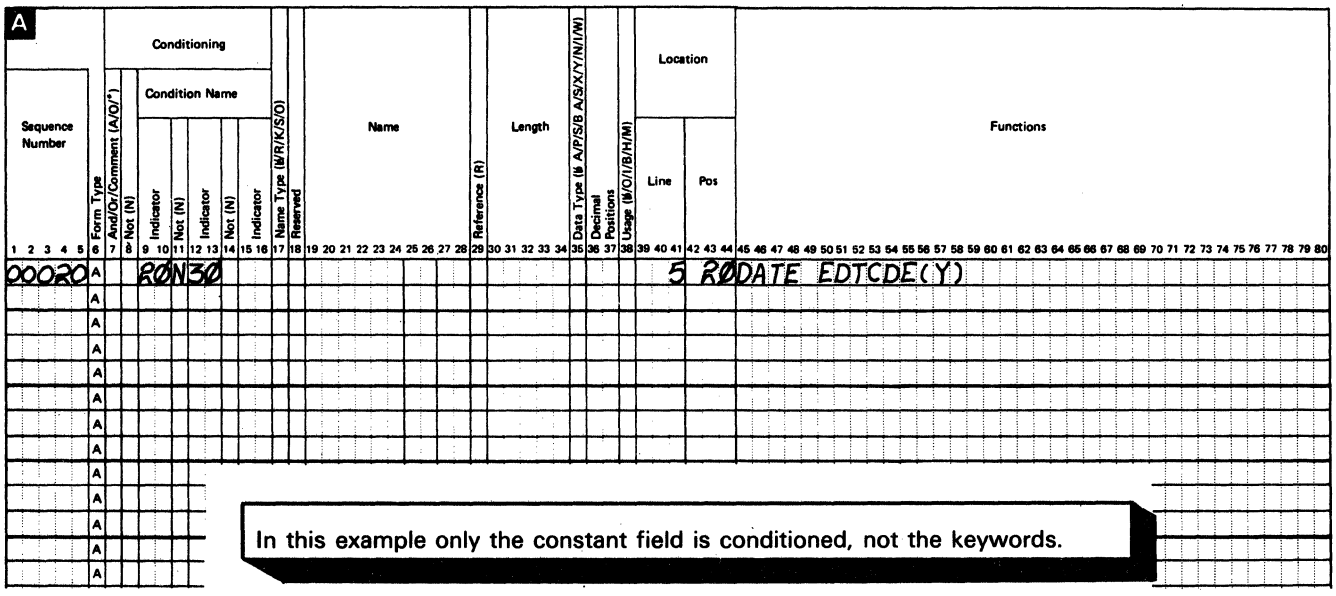
You can optionally specify an edit code or an edit word for this field (see the EDTCDE and EDTWRD keywords). If no edit code or edit word keyword is specified, no editing takes place.

Option indicators are not valid for this keyword.

Notes:

1. The Y edit code uses the character in the system value QDATSEP, as the date separator character. When DATE and EDTCDE(Y) are specified together on a constant (unnamed) field, the data is edited according to the system values QDATSEP (date separator character) and QDATFMT (date format). See the description of the EDTCDE keyword for more information on how the date is edited.
2. At execution time, if QDATFMT is JUL (Julian), and any edit code is specified, CPF puts a zero in the sixth position. EDTCDE(Y) should be used.

The following example shows how to specify the DATE keyword:



DFT('literal') | 'literal'

Use this field level keyword to specify the constant value for constant fields (unnamed fields) and to specify a default value for named fields.

Constant Fields:

The value of a constant field can be specified as a literal enclosed by apostrophes. (For other ways to specify a constant field, see the DATE and TIME keywords.) You can omit the DFT keyword itself, as well as the parentheses, to simplify the DDS. Whether you specify the DFT keyword explicitly or implicitly, CPF displays the specified literal as a constant field on the screen. See *Name (Positions 19 through 28)* earlier in this chapter for a description of constant fields.

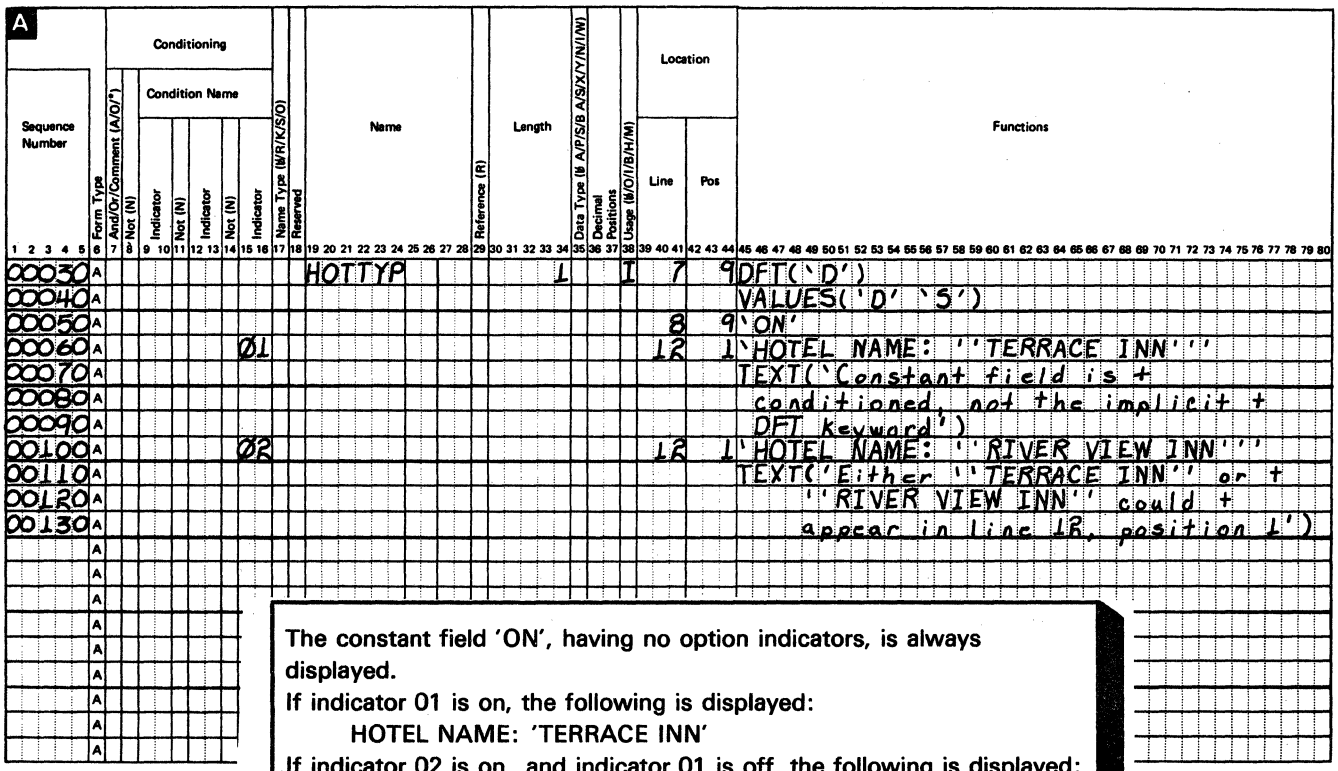
Named Fields:

For input-only fields, the specified literal is displayed when the field is displayed. The displayed value can then be changed by the work station user and returned to your program. For output-only, output/input, and message fields, you must also specify PUTOVR at the record level and OVRDTA at the field level with the DFT keyword. (For more information and an example, see the PUTOVR keyword.)

The EDTCDE and EDTWRD keywords cannot be specified with the DFT keyword.

Option indicators are not valid with this keyword, although option indicators can be used to condition the field (whether constant or named) with which it is specified.

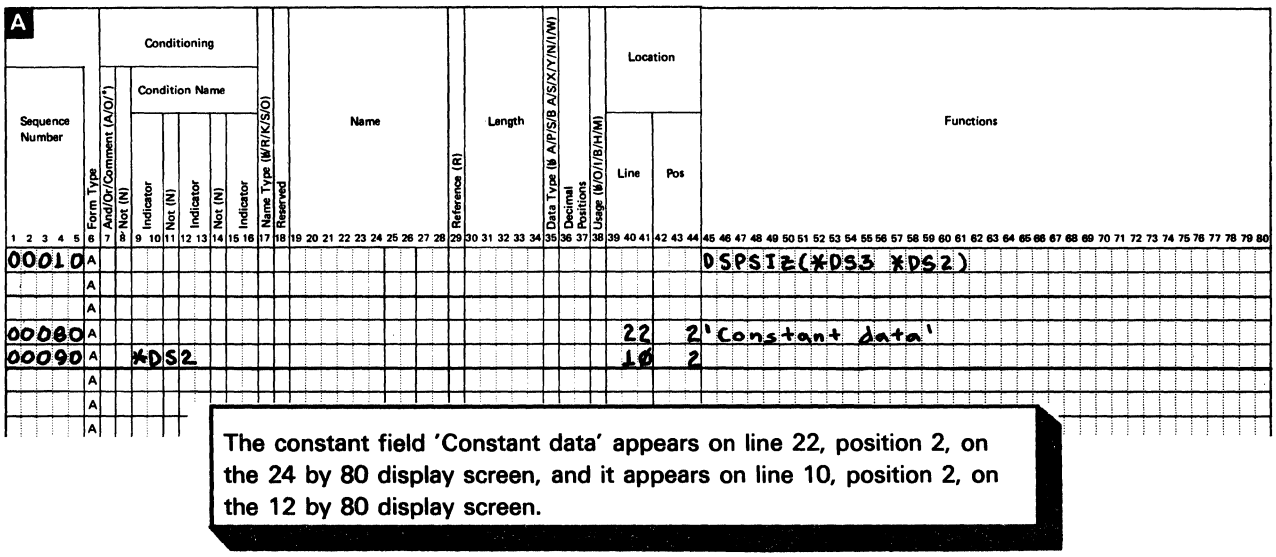
The following example shows how to specify the DFT keyword:



Sequence Number	Form Type	And/Or/Comment (A/O/*)	Or (N)	Conditioning: Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/N)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Pos	Functions
00030	A										HOTTYP		1			I	7	9	DFT('D')
00040	A																		VALUES('D' '5')
00050	A																8	9	'ON'
00060	A					01											12	1	'HOTEL NAME: ''TERRACE INN'''
00070	A																		TEXT('Constant field is +
00080	A																		conditioned, not the implicit +
00090	A																		DFT keyword')
00100	A					02											12	1	'HOTEL NAME: ''RIVER VIEW INN'''
00110	A																		TEXT('Either ''TERRACE INN'' or +
00120	A																		''RIVER VIEW INN'' could +
00130	A																		appear in line 12, position 1')

The constant field 'ON', having no option indicators, is always displayed.
If indicator 01 is on, the following is displayed:
HOTEL NAME: 'TERRACE INN'
If indicator 02 is on, and indicator 01 is off, the following is displayed:
HOTEL NAME: 'RIVER VIEW INN'

If you are specifying a constant field for more than one screen size, and you are not changing the contents of the field for the different screen sizes, only the location of the field, then do not repeat the literal. The following example shows how to do this:



Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (W/O/I/B/H/M)	Location: Line	Pos	Functions
00010	A																		DSPSIZ(*DS3 *DS2)
	A																		
	A																		
00080	A																22	2	'Constant data'
00090	A			*DS2													10	2	
	A																		
	A																		
	A																		

The constant field 'Constant data' appears on line 22, position 2, on the 24 by 80 display screen, and it appears on line 10, position 2, on the 12 by 80 display screen.

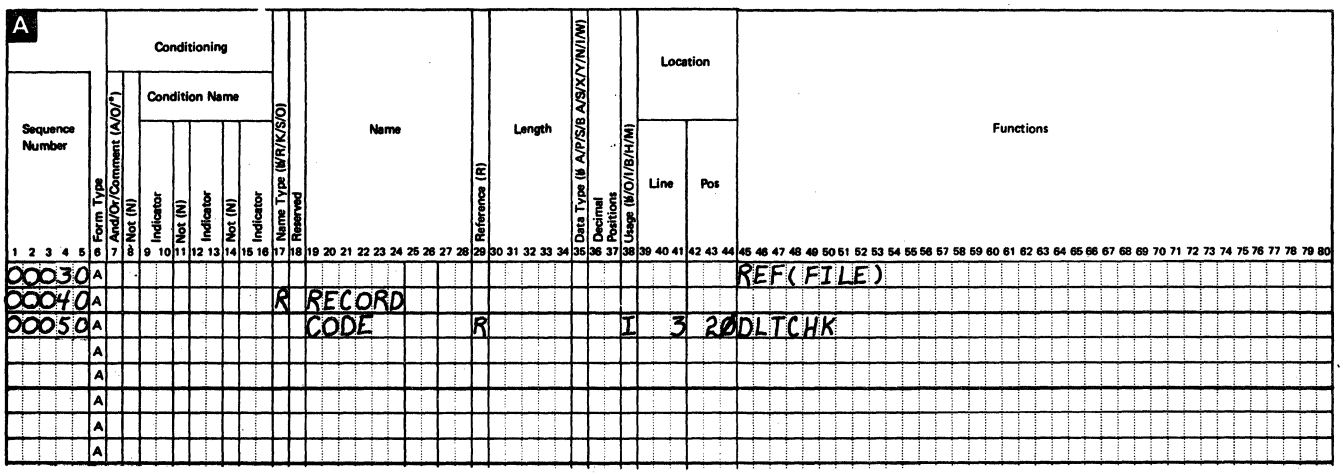
DLTCHK

Use this field level keyword to specify that CPF is to ignore all validity checking keywords that are specified for a referenced field. This keyword is valid only when R is specified in position 29.

If you specify *any* new validity checking keywords, DLTCHK is unnecessary; the new validity checking keywords override the referenced validity checking keywords.

Option indicators are not valid with this keyword.

The following example shows how to specify the DLTCHK keyword:



DLTEDT

Use this field level keyword to specify that CPF is to ignore the EDTCDE or EDTWRD keyword if one of them is specified for a referenced field. This keyword is valid only when R is specified in position 29.

If you specify a new editing keyword, DLTEDT is unnecessary; the new editing keyword overrides the referenced editing keyword.

Option indicators are not valid with this keyword.

The following example shows how to specify the DLTEDT keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9, 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00040	A																		REF(FILEA)
00050	A								R		RECORD								
00060	A										AMT	R				B	5	20	DLTEDT
	A																		
	A																		
	A																		
	A																		
	A																		

DSPATR(attribute-1 [attribute-2 [attribute-3 [...]]])

Use this field level keyword to specify one or more display attributes for the field you are defining. You can specify the DSPATR keyword more than once for the same field, and you can specify more than one attribute for the same keyword. However, each attribute (for example, UL), can be specified only once per field. The following are valid attributes:

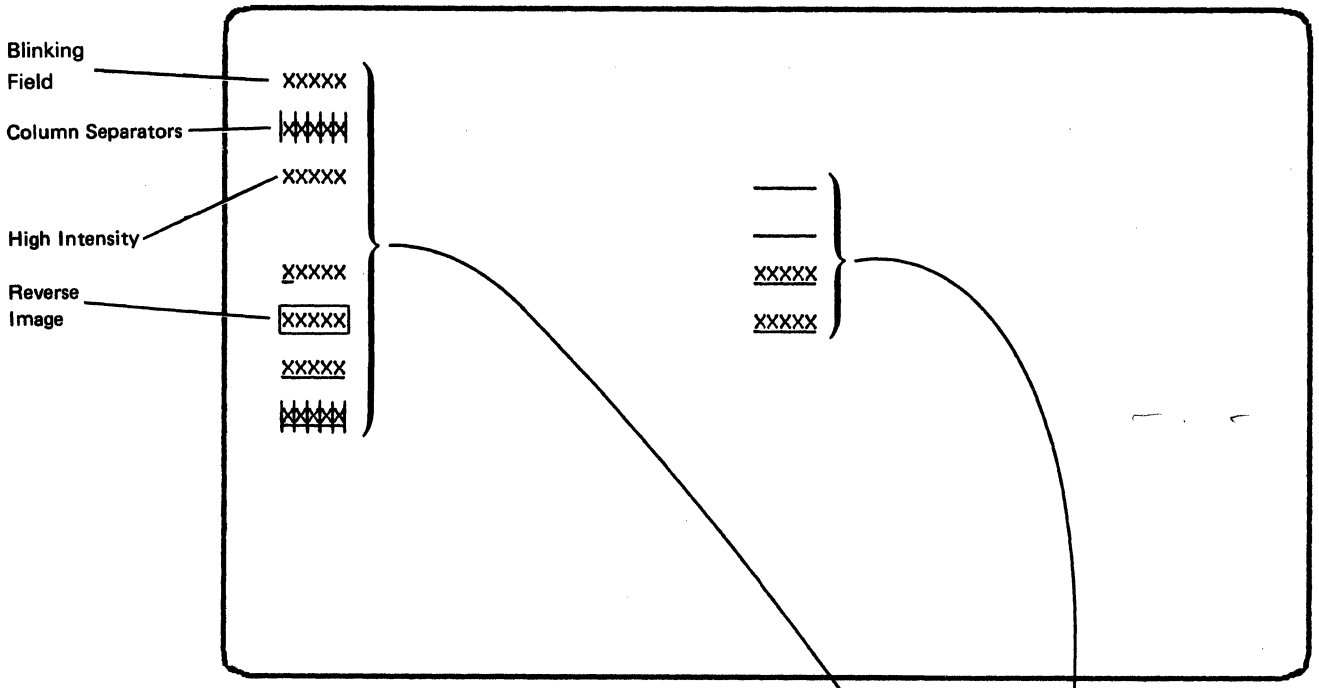
Display Attribute	Meaning
	For All Fields
BL	Blinking field
CS	Column separator
HI	High intensity
ND	Nondisplay
PC	Position cursor
RI	Reverse image
UL	Underline
	For Input-Capable Fields Only
MDT	Set modified data tag when displayed
OID	Operator identification
PR	Protect contents of field from input keying
SP	Select by light pen

Notes:

1. If you specify the UL, HI, and RI attributes on the 5250 work station for the same field, the result is the same as if you had specified ND.
2. If OID is specified, then SP should not be specifed. Neither OID nor SP can be optioned unless specified with another display attribute.

Option indicators are valid with this keyword, except when the attributes OID or SP are the only display attributes specified.

Detailed descriptions of each of the attributes follow the coding example and sample display provided in Figure 4-5.



IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050* Printed in U.S.A.

File	Keying Instruction	Graphic	Description	Page of
Programmer / Date		Key		

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		RECORD1								
00020	A										FLD1		5				2		2DSPATR(BL)
00030	A										FLD2		5				4		2DSPATR(CS)
00040	A										FLD3		5				6		2DSPATR(HI)
00050	A										FLD4		5				8		2DSPATR(ND)
00060	A										FLD5		5				10		2DSPATR(PC)
00070	A										FLD6		5				12		2DSPATR(RI)
00080	A										FLD7		5				14		2DSPATR(UL)
00090	A										FLD8		5				16		2DSPATR(UL CS)
	A																		
00095	A	X	Usage I or B at programmer's choice																
00100	A										FLD9		5			I	6	40	DSPATR(MDT)
00110	A										FLD10		5			I	8	40	DSPATR(PR)
00120	A										FLD11		5			B	10	40	DSPATR(OID)
00130	A										FLD12		5			B	12	40	DSPATR(SP)
	A																		
	A																		
	A																		
	A																		
	A																		

*Number of sheets per pad may vary slightly.

Figure 4-5. A 5-byte Field Displayed with Various Display Attributes

A description of each of these attributes follows:

(BL) Blink: Use this attribute to specify that the field is to blink when it is displayed on the screen.

This attribute is valid only for the 5250 work station; it is ignored for the system console.

(CS) Column Separator: Use this attribute to specify that each position of the field is to be displayed with a vertical bar at its left and right edge. When specified for a nondisplay field, the separators are displayed even though there are no characters between them. You can use column separators to precisely indicate cursor positioning within a field and to indicate the length of an otherwise blank field.

This attribute is valid only for the 5250 work station; it is ignored for the system console.

(HI) High Intensity: Use this attribute to specify that the field is to be intensified (highlighted) when it is displayed on the screen.

This attribute is valid only for the 5250 work station; it is ignored for the system console.

(ND) Nondisplay: Use this attribute to specify that the field is not to be displayed on the screen; the screen positions for this field appear blank. The attribute can be used for passwords or other security-sensitive data. If the print function (permitted by specifying the PRINT keyword) is performed, nondisplay fields are not printed.

This attribute is valid for the system console and the 5250 work station.

(PC) Position Cursor: Use this attribute to position the cursor to the first character position of the field you are defining. You can specify this attribute for several fields, and the cursor will be positioned at the first selected field with this attribute. Note that the fields within a record are ordered in line/position sequence as they appear on the screen and not necessarily in the order you specify them.

For a detailed discussion of cursor positioning, see the *CPF Programmer's Guide*.

This attribute is valid for the system console and the 5250 work station.

(RI) Reverse Image: Use this attribute to specify that the image of the field is to be reversed from the rest of the screen when it is displayed. Whether the screen is light-on-dark or dark-on-light depends on the state of the screen prior to displaying the field; this setting is controlled by the work station user.

This attribute is valid only for the 5250 work station; it is ignored for the system console.

(UL) Underline: Use this attribute to specify that the field is to be underlined when it is displayed on the screen. All input-capable fields are underlined by default; see the CHGINPDFT keyword to prevent the default underlining. (If CHGINPDFT is specified, DSPATR(UL) must be specified to underline an input-capable field.) If DSPATR(UL) is specified with option indicators and the option indicators are not satisfied (DSPATR(UL) not selected), the field appears without underline.

This attribute is valid for the system console and the 5250 work station.

(MDT) Set Modified Data Tag: Use this attribute to specify that CPF is to set on the modified data tag (MDT) for the field you are defining when the field is written to the display. The attribute ensures that the field is sent from the device when the record is read from the display. On a subfile record format, option indicators are valid for this attribute only if the value specified for SFLPAG equals the value specified for SFLSIZ (see the SFLPAG keyword description).

This attribute is valid for the system console and the 5250 work station.

Note: CPF saves output data for output/input fields or initialized data for fields with the DFT keyword specified. This causes the saved data to be returned on an input operation if no new (modified) data is keyed into the field.

This attribute is valid for input-capable fields and is not valid for output-only fields or unnamed constant fields.

(OID) Operator Identification: Use this attribute to specify that CPF is to allow magnetic stripe reader OID data to be entered into this field. If it is to be a nondisplay field also, the DSPATR(ND) attribute must be specified.

Magnetic stripe data can be entered into any input field. However, if the data is operator identification, this attribute *must* be specified. If no magnetic stripe reader exists at the work station in use, this keyword is ignored. The same field can be specified as a light-pen-selectable field, by means of DSPATR(SP), and as an operator identification field, by means of DSPATR(OID). However, if both of these attributes are specified, DSPATR(SP) is ignored. Because option indicators are valid for neither keyword, they should not be specified on the same field.

This attribute is valid only for the 5250 work station; it is ignored for the system console.

(PR) Protect: Use this attribute to specify that the work station user cannot key into the input-capable field you are defining. This attribute is valid for input-capable fields only; output-only fields and constant fields are protected by definition. On a subfile record format, option indicators are valid for this attribute only if the value specified for SFLPAG equals the value specified for SFLSIZ (see the SFLPAG keyword description).

This attribute is valid for the system console and the 5250 work station.

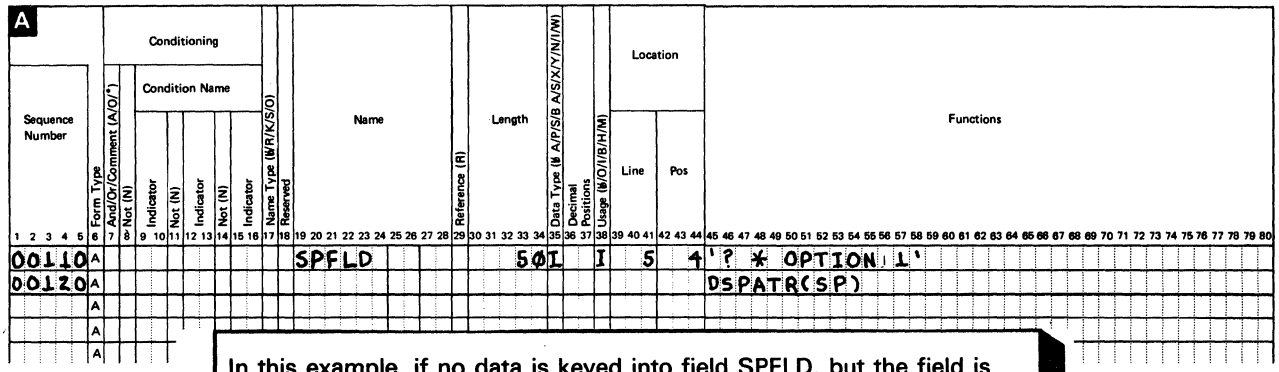
(SP) Select By Light Pen: Use this attribute to specify that this field is a light-pen-selectable field. The work station user can key into a light pen field unless an I (Inhibit keyboard entry) has been specified in position 35 (Data Type/Keyboard Shift) for the field.

When the field is first displayed, the contents of the field are set by your program (output/input field) or in the DDS (input-only field with DFT keyword or literal). If no new data is entered by the work station user, this output data is returned to your program on an input operation.

The recommended contents of this field are:

- A switch character, either hex 6F (?) or, if the work station user selects the field by a light pen, hex 6E (>)
- A blank (hex 40)
- A target character, which can be any character, such as an asterisk (*)
- Another blank
- Additional data to identify the field to the work station user (1 or more characters)

The following example shows how to specify the DSPATR(SP) keyword with an input-only field (showing the recommended data contents as a literal):



A	Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (M A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (M/O/I/B/H/M)	Location: Line	Location: Pos	Functions
	1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	00110	A										SPFLD		50	I		I	5	4	'? * OPTION 1'
	00120	A																		DSPATR(SP)
		A																		
		A																		
		A																		

In this example, if no data is keyed into field SPFLD, but the field is selected with the light pen, the data returned in field SPFLD would be:

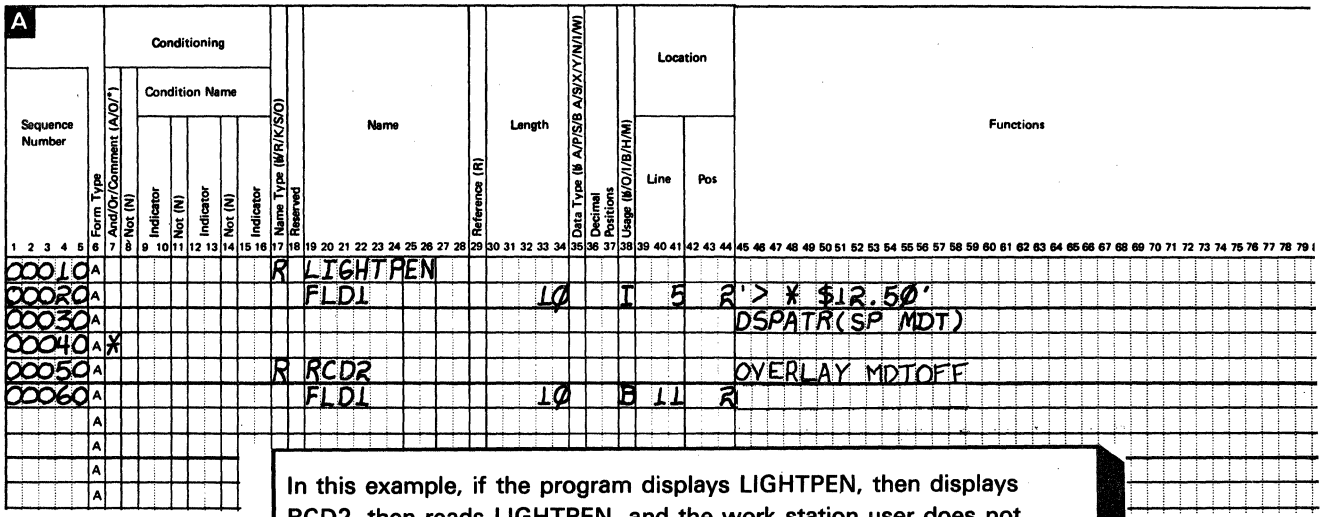
>b*bOPTIONb1

where the b character represents a blank.

When the work station user selects a field with the light pen, both the MDT bit and the first character of that field are changed. When the field is selected, the MDT bit is set on, changing the first character of the field to >. If the same field is selected again, the MDT bit is set off and the first character becomes ?.

By specifying a switch character, your program prevents the first character of data from being changed to > or ? when the field is selected by the light pen. If the MDT bit is on when your program issues an input operation to the record format, the contents of the field are returned to your program as a user-modified field.

If you use DSPATR(MDT) to set on the MDT of a light-pen-selectable field, then you should either omit the MDTOFF keyword from other record formats or you should read the light-pen-selectable field before displaying any record format with MDTOFF in effect. The following example shows what could happen:



A	Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (W/O/I/B/H/M)	Location: Line	Pos	Functions
	1 2 3 4 5	6	7	8	9 10	11 12	13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
	00010	A								R		LIGHTPEN								
	00020	A										FLD1		10			I	5		R'> * $12.50'
	00030	A																		DSPATR(SP MDT)
	00040	A	*																	
	00050	A								R		RCD2								OVERLAY MDTOFF
	00060	A										FLD1		10			B	11		R
		A																		
		A																		
		A																		
		A																		

In this example, if the program displays LIGHTPEN, then displays RCD2, then reads LIGHTPEN, and the work station user does not select FLD1 with the light pen, the MDT of FLD1 is turned off by the display of RCD2.

Also, the switch character of FLD1 is returned as ?, even though the field was not selected, and the switch character appears as >. The MDT and the switch character are in opposing states.

Notes:

1. This attribute is useful only for the 5250 work station with light pens and is ignored for the system console.
2. This attribute is only valid for input-capable fields.

DSPSIZ(DSx[*DSy[*DSz]])
DSPSIZ(lines chars [condition-name-1][lines chars[condition-name-2] [lines chars[condition-name-3]]])

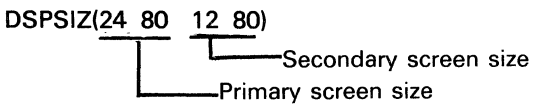
Use this file level keyword to specify the display size to which your program can open this display file. The DSPSIZ keyword is optional; if you do not specify it for a display file, the display file can be opened only to display devices with a 24 by 80 screen size. You can specify this keyword in one of two ways:

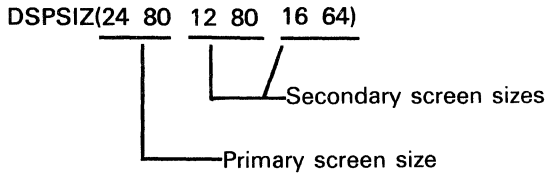
- *Using IBM-supplied screen size condition names*: Specify up to three parameter values as *DS1, *DS2, or *DS3, in any order. At least one parameter value is required; you cannot specify a parameter value twice. If you specify more than one parameter value, see *Primary and Secondary Screen Sizes* later in this keyword description.

- *Specifying lines and characters to permit user-defined screen size condition names*: Instead of the IBM-supplied screen size condition names, specify the screen size in lines and characters (only 16 64, 12 80, and 24 80 are valid). See Example 2 later in this keyword description. Optionally, you can also define a screen size condition name other than *DS1, *DS2, or *DS3. The screen size condition name you define must be from 2 to 8 characters long, and the first character must be an asterisk (*). You can specify these user-defined condition names in positions 7 through 16 (conditioning) on subsequent DDS statements at the field level. If you do not specify user-defined screen size condition names, you must use IBM-supplied screen size condition names to condition the location of fields.

Primary and Secondary Screen Sizes

Whether you use IBM-supplied screen size condition names or specify lines and characters directly, the first screen size you specify is the *primary screen size*. The second and third screen sizes, if specified, are *secondary screen sizes*). For example:





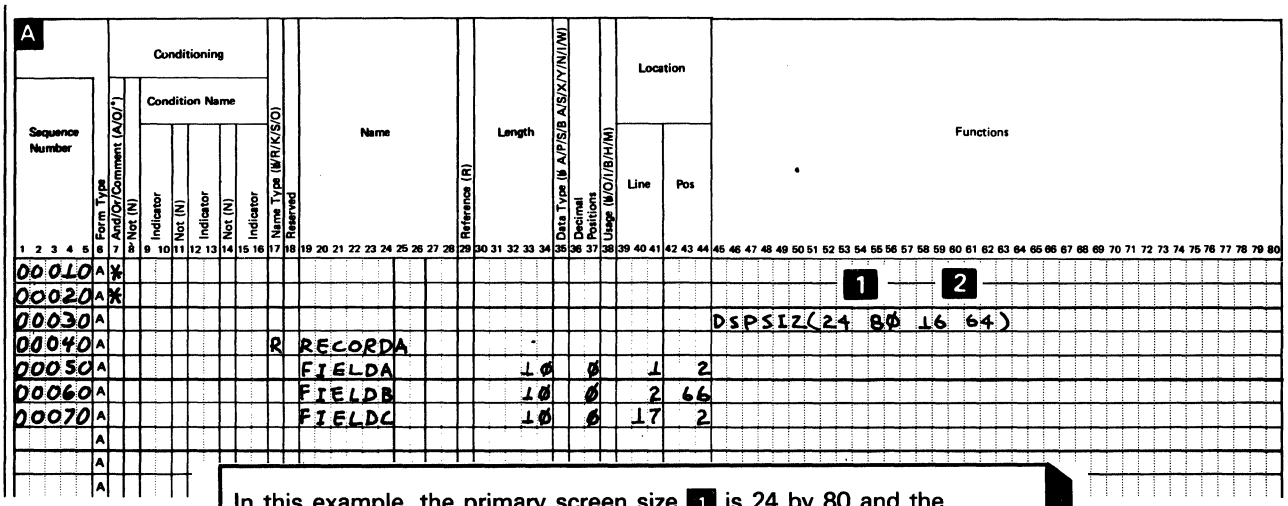
When you specify more than one screen size for the DSPSIZ keyword, you can specify screen size condition names in positions 7 through 16 on subsequent DDS statements at the record and field levels. These screen size condition names are then used to condition keywords and the locations of fields. See *Positions 7-16 (Conditioning)* in the *Positional Entries* section earlier in this chapter. Note that if you specify user-defined screen size condition names for the DSPSIZ keyword, you cannot use IBM-supplied screen size condition names for conditioning.

Valid Display Sizes	Display Device	Meaning
*DS1 or 16 64	System console	16 lines by 64 characters 1024 characters total
*DS2 or 12 80	5251 (Models 1 and 2), 5252	12 lines by 80 characters 960 characters total
*DS3 or 24 80	5251 (Models 11 and 12)	24 lines by 80 characters 1920 characters total

Option indicators are not valid with this keyword.

The following examples show how to specify the DSPSIZ keyword and the screen size condition names.

Example 1:



Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: And (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/N)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A	*																			
00020	A	*																			1 2
00030	A																				DSPSIZ(24 80 16 64)
00040	A										R		RECORDA								
00050	A												FIELDA		10		0		1	2	
00060	A												FIELDB		10		0		2	66	
00070	A												FIELDC		10		0		17	2	
	A																				
	A																				
	A																				

In this example, the primary screen size 1 is 24 by 80 and the secondary screen size 2 is 16 by 64. FIELDB is beyond position 64 and FIELDC is beyond line 16, so the data description processor gives them a location of *NOLOC in the expanded source listing for secondary screen size 16 by 64.

If the data description processor assigns *NOLOC to an input-capable field, that field is processed at execution time to the point of setting up the input buffer data to be returned in the user's input buffer. The field itself is not displayed; the work station user cannot enter into or modify these fields. No processing of any kind is done for output-only fields.

The following is a compiler listing for example 1:

```
5714SS1 R03 M00 811204             DATA DESCRIPTION            DSPSIZ1.SC217806       08/07/81 15:45:56        PAGE    1
File name -        DSPSIZ1.SC217806       Type of file -     DISPLAY
Source file -      QDDSSRC.SC217806       Member -           DSPSIZ1        08/07/81  15:03:40
Type of data -     *DATA
Options -          *SRC    *LIST
Authority -        *NORMAL
Text -             Sample listing for DDS manual
Compiler -         IBM System/38 Data Description Processor
                                       DATA DESCRIPTION SOURCE
SEQNBR  *... ... 1 ... ... 2 ... ... 3 ... ... 4 ... ... 5 ... ... 6 ... ... 7 ... ... 8  DATE
   100       A                                    DSPSIZ(24 80 16 64)
   200       A            R RECORDA
   300       A              FIELDA       10  0    1  2
   400       A              FIELDB       10  0    1 66
   500       A              FIELDC       10  0   17  1
                          * * * * *   E N D   O F   S O U R C E   * * * * *
```

```
5714SS1 R03 M00 811204             DATA DESCRIPTION            DSPSIZ1.SC217806       08/07/81 15:45:56        PAGE    2
                                        EXPANDED SOURCE
                                                                                       FIELD  BUFFER POSITION
SEQNBR  *... ... 1 ... ... 2 ... ... 3 ... ... 4 ... ... 5 ... ... 6 ... ... 7 ... ... 8   LEN      OUT     IN
   100   '                                        DSPSIZ(24 80 16 64)
   200                    R RECORDA
   300                      FIELDA       10S 00   1  2                                  10        1
   400                      FIELDB       10S 00   1 66                                  10       11
   400       * *DS1                              *NOLOC
   500                      FIELDC       10S 00  17  1                                  10       21
   500       * *DS1                              *NOLOC
                    * * * * *   E N D   O F   E X P A N D E D   S O U R C E   * * * * *
```

FIELDB and FIELDC have no location on *DS1, which corresponds to 16 64.

```
5714SS1 R03 M00 811204             DATA DESCRIPTION            DSPSIZ1.SC217806       08/07/81 15:45:56        PAGE    3
                                            MESSAGES
  MSGID     SEV   NUMBER    TEXT

                                         MESSAGE SUMMARY
   TOTAL     INFO(0-9)       WARNING(10-19)     ERROR(20-29)      SEVERE(30-99)
     0           0                  0                 0                 0
* CPF7301   00    File DSPSIZ1 created in library SC217806.
```

Example 2:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/K/Y/N/I/M)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	
00010	A																		DSPSIZ(24 80 *LARGE 16 64 *CON)	
00020	A									R		RECORDA								
00030	A											FIELDA		10	B	0		1	2	
00040	A											FIELDB		10	B	0		1	65	
00050	A		*CON															1	49	
00060	A											FIELDC		10	B	0		17	1	
00070	A		*CON															15	1	
	A																			
	A																			
	A																			
	A																			
	A																			

This example is similar to example 1, except that valid secondary locations are specified for FIELDB (line 1, position 49) and for FIELDC (line 15, position 1) on the secondary screen size (user-defined as *CON).

Example 3:

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050*
Printed in U.S.A.

File		Keying Instruction	Graphic								
Programmer	Date		Key								

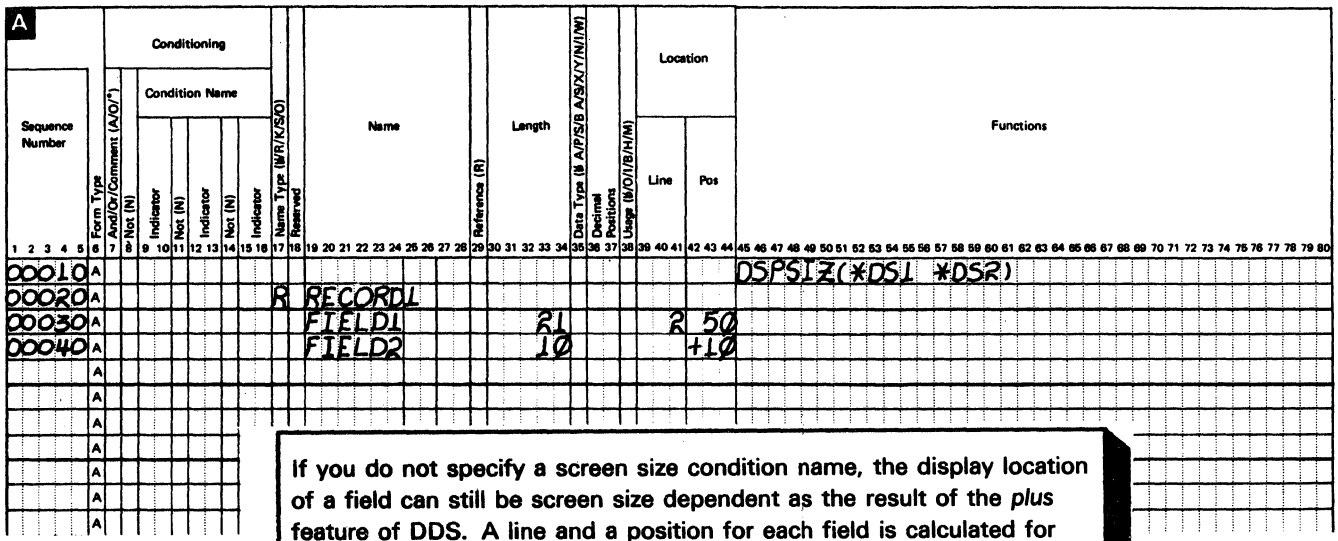
Description	Page	of

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A																		DSPSIZ(16 64 12 80 24 80)
00020	A								R		RECORDA								
00030	A										FIELD1						15	2	
00040	A			*DS2													11	2	
00050	A			*DS3													23	2	
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

This example shows how to reposition a field when the file is opened to different screen sizes. FIELD1 has valid locations on all screen sizes; it appears on the next to the last line on each screen size.

Example 4:



Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B/A/S/X/Y/N/I/M)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A																		DSPSIZ(*DS1 *DS2)
00020	A								R		RECORD1								
00030	A										FIELD1		21				2	50	
00040	A										FIELD2		10					+10	
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

If you do not specify a screen size condition name, the display location of a field can still be screen size dependent as the result of the *plus* feature of DDS. A line and a position for each field is calculated for each display size specified on the DSPSIZ keyword. If the plus value extends the field location beyond position 64, the field location is dependent on the screen size.

The following is a compiler listing for example 4:

```
5714SS1 R03 M00 811204                DATA DESCRIPTION              DSPSIZ4.SC217806        08/07/81 15:45:37          PAGE    1
File name -           DSPSIZ4.SC217806    Type of file -     DISPLAY
Source file -         QDDSSRC.SC217806    Member -           DSPSIZ4        08/07/81   15:23:26
Type of data -        *DATA
Options -             *SRC    *LIST
Authority -           *NORMAL
Text -                Sample listing for DDS manual
Compiler -            IBM System/38 Data Description Processor
                                    DATA DESCRIPTION SOURCE
SEQNBR  *... ... 1 ... ... 2 ... ... 3 ... ... 4 ... ... 5 ... ... 6 ... ... 7 ... ... 8  DATE
   100        A                                         DSPSIZ(16 64 12 80)
   200        A                R RECORDA
   300        A  66              FIELD1        21  0   2 50
   400        A                  FIELD2        10  0    +10
                         * * * * *   E N D   O F   S O U R C E   * * * * *
```

```
5714SS1 R03 M00 811204                DATA DESCRIPTION              DSPSIZ4.SC217806        08/07/81 15:45:37          PAGE    2
                                       EXPANDED SOURCE
                                                                                              FIELD  BUFFER POSITION
SEQNBR  *... ... 1 ... ... 2 ... ... 3 ... ... 4 ... ... 5 ... ... 6 ... ... 7 ... ... 8   LEN     OUT      IN
   100                                                  DSPSIZ(16 64 12 80)
              * OPTION INDICATOR OUTPUT BUFFER POSITIONS -
              *  *IN66 0001
   200'                        R RECORDA
   300           66              FIELD1        21S 00  2 50                                     21       2
   400                           FIELD2        10S 00  3 18                                     10      23
   400           *DS2                                  3  2
               * * * * *   E N D   O F   E X P A N D E D   S O U R C E   * * * * *
```

FIELD1 has same location on *DS1 and *DS2

FIELD2 has two locations:
- Line 3, position 18 on *DS1
- Line 3, position 2 on *DS2

```
5714SS1 R03 M00 811204                DATA DESCRIPTION              DSPSIZ4.SC217806        08/07/81 15:45:37          PAGE    3
                                          MESSAGES
 MSGID     SEV  NUMBER    TEXT

                                       MESSAGE SUMMARY
  TOTAL     INFO(0-9)      WARNING(10-19)     ERROR(20-29)     SEVERE(30-99)
     0          0                0                 0                 0
* CPF7301    00    File DSPSIZ4 created in library SC217806.
```

Special Cases: The following are descriptions of special cases you might encounter when specifying the DSPSIZ keyword:

- *DSPSIZ(*DS1 *DS3)*: All field locations for screen size *DS3 are the same as for screen size *DS1.

- All fields of a record can be described such that none fits on the screen size to which the file is opened. In this case, no fields are displayed. The record is handled as it would be for a larger screen where the fields fit. The record remains active until it is erased or overlaid. Active records can be read by your program; the input request is sent to the display device, and the work station user must respond to satisfy the request.

- All fields of a subfile record must fit within the specified subfile page, and the complete page must always fit (vertically) on the screen size on which it is displayed at execution time. Specify valid screen sizes by conditioning the SFLPAG keyword with screen size condition names. (Note that fields with display locations between 65 and 80 can still be specified but are not displayed on the system console.) See example 17 in Appendix F, *Examples*, for a subfile that can be displayed on two screen sizes.

- The following records occupy no screen space:
  - Records with no fields defined (this is different from none selected)
  - Records with only hidden or message fields
  - Records that have the CLRL keyword specified and that have no input-capable fields (these records can remain on the screen, but are not recognized by CPF for input operations or be erased through the use of the ERASE keyword)

  For implementation and programming purposes, these records are assumed to be located at 00 (from line 0 to line 0). On an output operation, any record located at 00 overlays a record at that location. When an overlap occurs, the previous record is disregarded and no longer considered active. The new record at location 00 is active and can be read by your program.

- If two fields in a record format have the same display location (line/position), they are treated as overlapping fields. Overlapping fields are not displayed at execution time. CPF checks each field as it is processed to ensure that it does not overlap a previously processed field.

  If a field does overlap, it is treated as an optioned field and not selected. To allow this execution-time checking, data description specifications must ensure all fields within a record are in primary location sequence, even when condition names are specified. For example, assume only one input field is specified for a record format and, according to the field location specification, this field overlaps a preceding output field. The work station user cannot enter any data because the input field is never displayed.

Note: The primary location sequence as it is seen in the display file must not be changed by specifying a different location sequence for secondary screen sizes. (A severe error occurs, and the file is not created.)

DUP[(response-indicator ['text'])]

Use this field level keyword to specify that the work station user can press the Dup key on the 5250 work station keyboard. The work station user presses the Dup key when the cursor is in this input-capable field to indicate that data for this field is to be duplicated from the record received on the previous input operation.

The actual duplicating function is the responsibility of your program. This keyword is valid for the 5250 work station; it is not valid for the system console.

When CPF recognizes that the user has pressed the Dup key (by detecting hex 1C in any field position), it handles the field as follows:

- If you have specified a response indicator or if the field is a character data type field, the data received from the display is returned to your program just as it was received. When the field is numeric, your program must be aware that arithmetic operations on the data passed to the program can cause an invalid data exception.

- If you do not specify a response indicator and the field is numeric, a value of zero (all hex F0s) is returned to your program.

- If you specify a response indicator and the field is numeric, the response indicator is set on when CPF detects hex 1C in the field. The indicator remains on until the record is returned to your program.

  When the Dup key is entered in a field, the indicator is set on. If a subsequent field fails validity checking (for instance RANGE keyword specified), CPF reissues a read to the display. If the user modifies the DUP field but does not press the Dup key, the DUP response indicator is returned to your program in the on condition, even though the input field no longer contains the DUP character.

- When CPF-detected input errors require additional input from the work station user, the response indicator for DUP remains on until the record is returned to your program.

Validity checking keywords (RANGE, VALUES, COMP, CHECK) can be specified with the DUP keyword; however, no field validation is done by CPF if the Dup key has been pressed.

The optional text is included on the listing generated at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The apostrophes are required. If you specify more than 50 characters between the apostrophes, the text is truncated to 50 characters on the program listing.

If specified on a subfile record format, option indicators are valid for the DUP keyword only if the value specified for the SFLPAG keyword is equal to the value specified for the SFLSIZ keyword (see the SFLPAG keyword description).

Option indicators are valid with this keyword.

The following example shows how to specify the DUP keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00020	A										FLDA	R				I	3	2	DUP(16 'FLDA was duped')
00030	A										FLDB		10			I	5	2	DUP
	A																		
	A																		
	A																		

EDTCDE(edit-code [* | floating-currency-symbol])

Use this field level keyword to specify an edit code for editing of output data for a *numeric-only* field (Y specified in position 35) using this current field description. An edit code is a required parameter value for this keyword.

EDTCDE is valid only for numeric-only fields. Both EDTCDE and EDTWRD keywords cannot be specified for the same field. You can use the EDTCDE keyword to perform edit functions such as the following:

- Suppress leading zeros
- Omit a sign from the low-order position of the field
- Punctuate the field without having to specify an edit word (see the EDTWRD keyword description)
- Perform asterisk fill
- Specify floating currency symbol (this symbol must correspond to the system value QCURSYM)
- Perform user-defined edit functions

Option indicators are not valid with this keyword.

CPF Edit Codes: The edit codes that you can specify as a parameter value for this keyword are:

1 through 4
A through D
J through M
Y and Z

Edit code X has no effect on display files. For an explanation of its use, see the EDTCDE keyword descriptions in Chapter 2, *Physical Files* and Chapter 3, *Logical Files*.

The following chart summarizes the functions provided by CPF edit codes. A negative field can be punctuated with no sign, CR (credit), or a minus.

Edit Code	Commas[1] Displayed	Decimal Points[1] Displayed	Sign Displayed When Negative Number			Zero Balance Displayed	Leading Zero Suppressed
			No Sign	CR	- (Minus)		
1	Yes	Yes	No sign			Yes	Yes
2	Yes	Yes	No sign				Yes
3		Yes	No sign			Yes	Yes
4		Yes	No sign				Yes
A	Yes	Yes		CR		Yes	Yes
B	Yes	Yes		CR			Yes
C		Yes		CR		Yes	Yes
D		Yes		CR			Yes
J	Yes	Yes			–	Yes	Yes
K	Yes	Yes			–		Yes
L		Yes			–	Yes	Yes
M		Yes			–		Yes
Y[2]							Yes
Z[3]							Yes

[1]The QDECFMT system value determines the decimal point character (period in U.S. usage), the character used to separate groups of three digits (comma in U.S. usage), and the type of zero suppression (depending on comma and period placement). See the *CPF Programmer's Guide* for detailed instructions.

[2]The Y edit code suppresses the leftmost zero of a date field that is three to six digits long, and it suppresses the two leftmost zeros of a field that is seven positions long. The Y edit code also inserts slashes (/) between the month, day, and year according to the following pattern:

nn/n
nn/nn
n/nn/n
nn/nn/nn
nnn/nn/nn

If the DATE keyword is specified with EDTCDE(Y), the separator character used is the system value, QDATSEP. The slash (/) is the default QDATSEP. If QDATFMT is JUL (Julian), the data is normally formatted as: nn/nnn.

[3]The Z edit code removes the sign (plus or minus) from a numeric field. The sign of the units position is changed to a hexadecimal F before the field is written.

User-Defined Edit Codes: You can also specify any of five user-defined edit codes (5 through 9). Before creating a display file from DDS source containing user-defined edit codes, the user-defined edit codes must exist. When creating the display file, the editing information is extracted from the user-defined edit codes as referenced in the DDS. Changing user-defined edit codes does not affect existing display files unless the display files are re-created. See the *CPF Programmer's Guide* and the *CL Reference Manual* for a description of how to create user-defined edit codes if those shipped with the system cannot be used.

Asterisk Fill and Floating Currency Symbol: You can optionally specify asterisk fill or floating currency symbol with edit codes 1 through 4, A through D, and J through M.

When you specify asterisk fill, an asterisk (*) is printed for each zero that is suppressed. A complete field of asterisks is printed for a zero balance field.

When you specify floating currency symbol, the symbol appears to the left of the first significant digit. The symbol does not print on a zero balance when an edit code is used that suppresses the zero balance. (The symbol that you specify must match the system value for the floating currency symbol. The symbol must match when the file is created; it does not have to match when the file is used.)

Note: If an edit code is changed after a file is created, the editing specified at the time the file was created is used. The new edit code is not used unless the file is re-created.

Figure 4-6 shows edit codes, unedited source data, and edited output. Zero suppression and decimal characters are determined by the system value QDECFMT. The date separator character is determined by the system value QDATSEP. In this figure, QDECFMT is assumed to equal b (blank), and QDATSEP is assumed to equal /.

Edit Codes	Positive Number–Two Decimal Positions	Positive Number–No Decimal Positions	Negative Number–Three Decimal Positions	Negative Number–No Decimal Positions	Zero Balance–Two Decimal Positions[1]	Zero Balance–No Decimal Positions[1]
Unedited[2]	1234567	1234567	ƀƀƀƀ125–	ƀƀƀƀ125–	ƀƀƀƀƀƀ	ƀƀƀƀƀƀ
1	12,345.67	1,234,567	.125	125	.00	0
2	12,345.67	1,234,567	.125	125		
3	12345.67	1234567	.125	125	.00	0
4	12345.67	1234567	.125	125		
A	12,345.67	1,234,567	.125CR	125CR	.00	0
B	12,345.67	1,234,567	.125CR	125CR		
C	12345.67	1234567	.125CR	125CR	.00	0
D	12345.67	1234567	.125CR	125CR		
J	12,345.67	1,234,567	.125–	125–	.00	0
K	12,345.67	1,234,567	.125–	125–		
L	12345.67	1234567	.125–	125–	.00	0
M	12345.67	1234567	.125–	125–		
Y[3]	123/45/67	123/45/67	0/01/25	0/01/25	0/00/00	0/00/00
Z[4]	1234567	1234567	125	125		

[1]The ƀ represents a blank.
[2]This is an example of a signed numeric field. For all other data types, the negative number would be displayed as 00012N whether decimal positions are specified or not, and the zero balanced field would appear as 000000 (whether decimal positions are specified or not).
[3]The Y edit code suppresses the leftmost zero of a date field that is three to six digits long, and it suppresses the two leftmost zeros of a field that is seven positions long. For more information, see the description of this edit code under *CPF Edit Codes* in this chapter.
[4]The Z edit code removes the sign (plus or minus) from a numeric field and suppresses leading zeros of a numeric field.

Figure 4-6. Valid Edit Codes, Source Data, and Edited Output

The following example shows how to specify the EDTCDE keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A										PRICE		5		2		1	10	EDTCDE(J)
00020	A										SALES		7		2		2	10	EDTCDE(K $)
00030	A										SALARY		8		2		3	10	EDTCDE(1 *)

In this example, the display length for PRICE is 7 because the J edit code is specified, causing the field to contain a decimal point and an ending minus sign. It is edited as:

ddd.dd-

where d represents a digit.

The display length for SALES is 11 because the K edit code and floating currency symbol are specified. It is edited as:

$dd,ddd.dd-

The display length for SALARY is 10 because the edit code 1 is specified with asterisk fill. It is edited as:

ddd,ddd.dd

EDTWRD('edit-word')

If you cannot accomplish the desired editing by using the EDTCDE keyword, you can specify an edit word instead. An edit word specifies the form in which the field values are to be displayed and clarifies the data by inserting characters directly, such as decimal points, commas, floating and fixed currency symbol, and credit balance indicators. The edit word can also be used to suppress leading zeros and to provide asterisk fill protection.

Option indicators are not valid with this keyword.

Use the following rules to specify a valid edit word:

- The EDTWRD keyword is valid for numeric only fields (Y specified in position 35).
- You cannot specify both EDTWRD and EDTCDE for the same field.
- Enclose the edit word in apostrophes.
- The sum of the blanks and stop-zero-suppression characters (digit positions) in the edit word must equal the length of the field.
- When using the floating currency symbol, the number of digit positions contained in the edit word must be equal to the number of positions in the field to be edited. The currency symbol is not counted as a digit position. For example, if you specify a field length of 7 with 2 decimal positions, the edit word must be specified as:

  EDTWRD('ƀ ƀ ƀ ƀ$0.ƀ ƀ')

  where the ƀ character represents a blank.

- Any displayable character is valid, but the following characters in certain positions have special meanings:

  – *Blank*: A blank is replaced with the character from the corresponding position of the data field. A blank position is referred to as a digit position.

  – *Ampersand*: An ampersand causes a blank in the edited field. The ampersand is not displayed. Note that ampersands specified in the edit word between blanks can result in invalid data when specified for an output/input field. This is because embedded blanks in a numeric-only field are converted to zeros. For example, if EDTWRD('ƀ ƀ&ƀ ƀ&ƀ ƀ&LATER') is specified for an output/input field, and the work station user changes the displayed field, CPF converts the first two ampersands to zeros and displays an error message (too many digits) when the work station user presses the Enter/Rec Adv key.

Initial Data in Program	Displayed Field	User Change to Field	Data after Conversion
060181	06 01 81 LATER	07 01 81 LATER	07001081

- *Zero*: A zero stops zero suppression; place it in the rightmost position where zero suppression is to stop. The zero is replaced with the character from the corresponding position of the data field, unless that character is a zero. Any zeros in the data that appear to the right of the stop-zero-suppression character are displayed. The stop-zero-suppression character is considered a digit position. At least one leading zero is suppressed. Each zero that is suppressed is replaced by a blank.

- *Asterisk*: An asterisk stops zero suppression and replaces zeros with asterisks (asterisk protection). Place the asterisk in the rightmost position where zero suppression is to stop. Each zero that is suppressed is replaced by an asterisk.

  An asterisk preceding a zero is interpreted as representing asterisk protection, and in this case, the zero prints as a constant. Any asterisks or zeros to the right of the stop-zero-suppression character are constants.

- *Currency Symbol*: A currency symbol coded immediately to the left of the zero suppression code causes the insertion of a currency symbol in the position to the left of the first significant digit. It is called the *floating currency symbol* when used in this manner.

  A currency symbol coded in the leftmost position of the edit word is fixed and prints in the same location each time. When used in this manner, it is called the *fixed currency symbol.*

  The currency symbol is not considered a digit replace position. This symbol must correspond to the system value QCURSYM.

- *Decimals* and *Commas*: Decimals and commas are printed in the same relative positions in which they are coded in the data unless they are to the left of the first significant digit. In that case, they are blanked out or replaced by an asterisk.

- *All Other Characters*: All other characters are printed if they are to the right of significant digits in the data field. If they are to the left of the high-order significant digits in the data word, they are blanked out or replaced by asterisks if asterisk protection is being used.

- If you want to show a negative sign with a negative number, include a sign in the edit word. Use either the minus sign (-) or the letters CR (credit). These print only if the number is negative.

Figure 4-7 shows sample edit words with the program value of the field and the display value of the field (as edited):

Edit Word	Program Value	Displayed As
' , , 0. &CR*'	0000000005-	.05 CR*
' , ,$0 . -*'	0000000005+	$0.05 *
' , , $0. CR**'	0034567890-	$345,678.90CR**
'$, , 0. '	0000000000	$ .00
'$& , , 0 . &-&GROSS'	1234567890-	$ 12,345,678.90 - GROSS
' , , * . &-'	0000135792	*****1,357.92
	0000135792	0000135792
	0000135792-	0000135792-
' '	0000000000	
' '	0000135678+	135678
' '	0000135678-	135678
' 0'	0000135678-	135678
'0 '	0000135678+	0000135678
'$ &-&NET'	0000135678+	$ 135678 NET
'$ &-&NET'	0000135678-	$ 135678 - NET
'$0 -&NET'	0000135678	$ 0000135678 NET
' $0 &CR*'	0000135678-	$135678 CR*
' $0 &CR*'	1234567809-	$12345678 09 CR*
' *&CR'	0000000000-	********** CR
' * &CR'	0000000000-	*********00 CR
'* '	0000135678-	*0000135678
' , , . &CR*&NET'	0000135678-	1,356.78 CR* NET
' , , . &CR*&NET'	0000135678	1,356.78 * NET
' , , $0. '	0000000005	$.05
' , , $0. CR'	0001356789-	$13,567.89CR
' , , *. *CR**'	0000135678+	*****1,356.78* **
' , , DOLLARS CENTS'	0000135678	1,356DOLLARS78CENTS
'0 - - '	095140036	95-14-0036
'& ,* 0, '	013579	**130,579
' - - &LATER'	093076	9-30-76 LATER
(A) ' & & &LATER'	093076	9 30 76 LATER
' / / '	100176	10/01/76

(A) **Note:** An execution time error can occur if this edit word is specified for a both field. See the description of ampersands earlier in this keyword description.

Figure 4-7. Sample Edit Words

The following example shows how to specify the EDTWRD keyword:

A

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00020	A										FIELDA		7		2		5	2	EDTWRD(' $0. ')
	A																		
	A																		
	A																		

ERASE(record-name-1 [record-name-2 ...[record-name-20]])

Use this record level keyword with the OVERLAY keyword to specify that the records whose names you supplied as parameter values are to be erased from the screen when this record is written. The ERASE keyword may be specified more than once.

The OVERLAY keyword must be specified when the ERASE keyword is specified.

If ERASE and CLRL are both in effect on an output operation, the records specified in the ERASE keyword are erased despite the CLRL keyword.

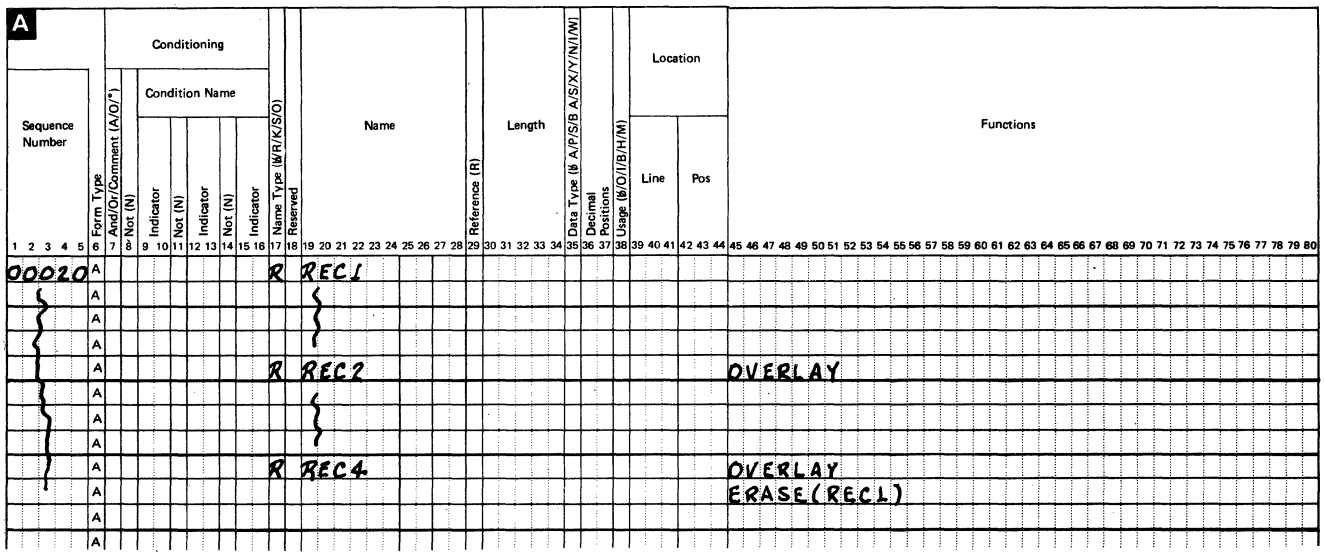
A record already on the display that has no input-capable fields and that has CLRL specified cannot be erased by the ERASE keyword on another record (the ERASE keyword has no effect).

If the specified record is not on the screen, this function is ignored for that record name.

Option indicators are valid with this keyword.

Note: This function requires extra data transmission and should be used with discretion.

The following example shows how to specify the ERASE keyword:



ERASEINP [{ ***MDTON** / ***ALL** }]

Use this record level keyword with the OVERLAY keyword to erase unprotected input-capable fields already on the screen. The fields are erased before the record format you are defining is displayed. Input-capable fields in the record format you are defining are not erased.

To erase all input-capable fields already on the screen, specify the *ALL parameter. To erase only input-capable fields that have their modified data tags (MDTs) set on, specify the *MDTON parameter. Specifying the ERASEINP(*MDTON) or the ERASEINP keyword has the same effect as pressing the Erase Input key on the 5250 work station keyboard.

On the system console, the ERASEINP(*MDTON) keyword is treated as the ERASEINP(*ALL) keyword.

The OVERLAY keyword must be specified when the ERASEINP keyword is specified.

Unprotected input-capable fields are fields for which the DSPATR(PR) keyword is not in effect.

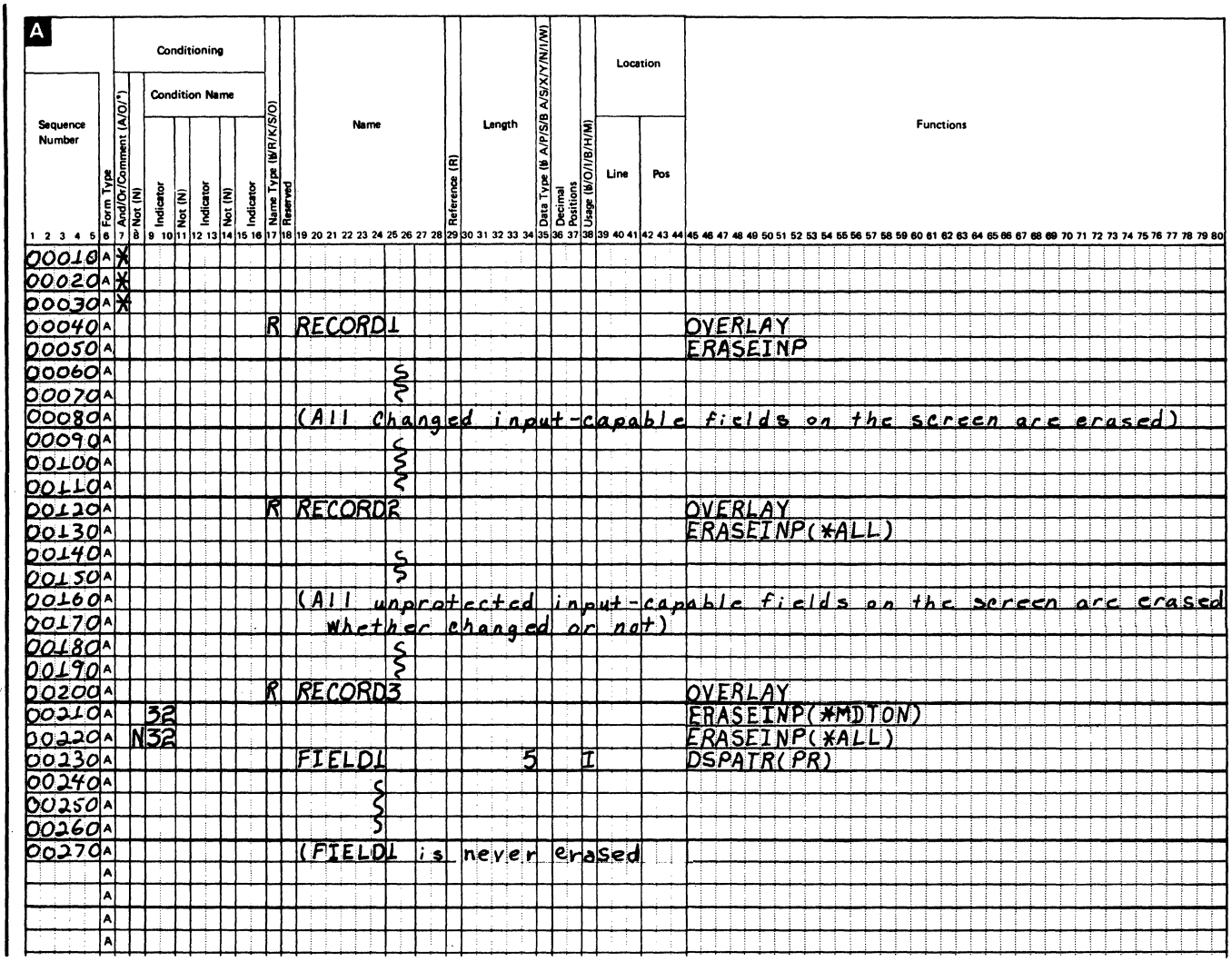
When the MDTOFF keyword is specified on the same record format as the ERASEINP keyword, two conditions can occur:

- ERASEINP(*ALL) implies MDTOFF(*UNPR) unless MDTOFF(*ALL) is specified.
- If ERASEINP or ERASEINP(*MDTON) is specified with MDTOFF(*ALL), the end effect is as if ERASEINP(*ALL) and MDTOFF(*ALL) are both specified.

If ERASEINP and PROTECT are both in effect for an output operation, CPF first erases the input-capable fields specified on the ERASEINP parameter value, then protects all input-capable fields on the display from input keying.

Option indicators are valid with this keyword.

The following example shows how to specify the ERASEINP keyword:



Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A	*																	
00020	A	*																	
00030	A	*																	
00040	A								R		RECORD1								OVERLAY
00050	A																		ERASEINP
00060	A										~								
00070	A																		
00080	A										(All changed input-capable fields on the screen are erased)								
00090	A										~								
00100	A										~								
00110	A										~								
00120	A								R		RECORD2								OVERLAY
00130	A																		ERASEINP(*ALL)
00140	A										~								
00150	A										~								
00160	A										(All unprotected input-capable fields on the screen are erased								
00170	A										whether changed or not)								
00180	A										~								
00190	A										~								
00200	A								R		RECORD3								OVERLAY
00210	A			32															ERASEINP(*MDTON)
00220	A		N	32															ERASEINP(*ALL)
00230	A										FIELD1		5			I			DSPATR(PR)
00240	A										~								
00250	A										~								
00260	A										~								
00270	A										(FIELD1 is never erased								
	A																		
	A																		
	A																		
	A																		

ERRMSG('message-text' [response-indicator])
ERRMSGID(msgid msg-file[.library-name])

Use one of these field level keywords to identify a message to be displayed on the message line and associated with this field.

ERRMSG Keyword: For ERRMSG, the parameters specify the message text and, optionally, a response indicator. The message text is the message to be displayed. (The Help key is not supported and if pressed causes an error message to be sent to the display.)

If you specify a response indicator, it should be the same as the option indicator used to condition the ERRMSG keyword. On the input operation that follows the display of the error message, CPF turns off the indicator. If the response and the option indicators are the same, they are both turned off. One exception to this rule is if the response indicator is also specified for another keyword such as CHANGE, CAnn, or CFnn. In that case, the on/off setting of the response indicator is based on the results of the function provided by the CHANGE or CFnn keyword. When a response indicator is specified, the first 50 characters of the message are also used as indicator text. Separate response indicator text is not valid for the ERRMSG keyword.

ERRMSGID Keyword: For ERRMSGID, the parameters specify the message identifier for the message to be displayed and its message file. The message to be displayed is retrieved from the message file during program execution. (The Help key is supported for this keyword and if pressed causes a predefined second-level message from the message file to be displayed if it exists.) The message file can optionally be qualified by a library name.

Note: The message file used in ERRMSGID must exist in the specified library. The work station user must have operational rights to the message file at program execution time.

The following example shows how to specify the ERRMSG and ERRMSGID keywords:

A

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00010	A								R		CUSMST								
00020	A										⎧								
00030	A										⎨								
00040	A										⎩								
00050	A										QTYORD	R				I	5	3	
00060	A			61															ERRMSG('No stock available' 61)
00070	A			62															ERRMSG('Partial stock available' +
00080	A																		62)
00090	A			63															ERRMSGID(MSG2000 CONSOLEMSG)

Priority Among Selected Keywords: You can specify ERRMSG and ERRMSGID more than once for a single field. During program execution, use option indicators to select a particular message to be displayed.

Only one message can be displayed at one time even if messages are in effect for several fields on the same output operation. The field whose message is displayed is the first field for which the program selected a message.

If several keywords are in effect for one field on an output operation, the message to be displayed is the first of the following:

1. ERRMSG (If more than one ERRMSG keyword is selected, the first one the program selects is displayed.)

2. ERRMSGID (If more than one ERRMSGID keyword is selected, the first one the program selects is displayed.)

A message field is displayed only if no error message keywords are also to be displayed.

For a list of priorities including SFLMSG and SFLMSGID, see *Priority Among Selected Keywords* in the SFLMSG/SFLMSGID keyword description later in this chapter.

Conditions Occurring During Message Display: The displaying of a message using ERRMSG and ERRMSGID is similar to the displaying of messages by CPF when field validation errors are detected.

When a message is displayed because of either the ERRMSG or the ERRMSGID keyword, all fields on the screen are retained, including the field the message is associated with. Except for option indicators, data in the output buffer is ignored (that is, any new data from the program is not sent to the display).

The command keys valid following display of a message are:

- Command keys specified at the file level.
- Command keys specified for the record format for which a message is displayed, if selected when the message is displayed.

When the message is displayed on the 5250 work station, the following conditions occur:

- For all errors:
  - The message is highlighted.
  - The cursor is blinked and the keyboard locked until the work station user presses the Reset key.
- For errors associated with input-capable fields:
  - All fields in error are displayed with their images reversed. (If a field in error has both the underline (UL) display attribute and the highlight attribute (HI), its image is not reversed.)
  - The cursor is repositioned to the first displayed field that is in error.
- For errors associated with output-only fields:
  - The display attribute of the field is not changed.
  - The cursor is not positioned to the field (it does not change position).

When the message is displayed on the system console, the following conditions occur:

- The cursor is repositioned to the associated field.
- The message appears without special display attributes.

Restoration of Reversed Image Fields: Fields on the 5250 work stations are displayed with their images reversed because of system-detected keying errors or because of the ERRMSG or the ERRMSGID keyword. Generally, CPF restores the image on the next I/O operation to the display, and that is usually the next request from your program. The restoration is done before the requested function is performed. The following are exceptions where requests from your program do not cause CPF to restore reversed image fields:

- An input request with cancel (canceling a get operation with NOWAIT)
- A close request when the KEEP keyword is in effect
- Any request to a subfile record (no data is sent to the device)
- An output operation to a subfile control record format that does not display the subfile control record or subfile records (for example, clearing, deleting, or initializing the subfile)

Priority Among Selected Keywords: For a description of which message is displayed when more than one error message or subfile message keyword is selected, see *Priority Among Selected Keywords* in the SFLMSG/SFLMSGID keyword description.

Restrictions and Notes

1. If ERRMSG or ERRMSGID is in effect, no record or field processing is performed other than that defined for these keywords. If neither keyword is in effect, the record is processed in the normal manner.
2. ERRMSG and ERRMSGID are valid for output-only, input-only, or output/input fields. ERRMSG and ERRMSGID cannot be specified for a constant, hidden, or message field.
3. For input-capable fields, ERRMSG and ERRMSGID are in effect only if the field for which they are specified is already on the screen.
4. ERRMSG and ERRMSGID cannot be specified in a subfile record format (SFL keyword specified). To display error messages for a subfile, see the SFLMSG and SFLMSGID keywords.
5. ERRMSG and ERRMSGID are ignored if the variable start line number (SLNO(*VAR) specified) has changed since the last output operation.

GETRETAIN

Use this record level keyword with the UNLOCK keyword to specify that CPF is not to erase input-capable fields on input operations as described under the UNLOCK keyword.

You must specify the UNLOCK keyword without any parameters before specifying the GETRETAIN keyword. The UNLOCK(*MDTOFF) specification provides the same function as the GETRETAIN keyword. See the UNLOCK keyword for additional details.

Option indicators are not valid with this keyword.

The following example shows how to specify the GETRETAIN keyword:

A

Sequence Number (1-5)	Form Type (6)	Conditioning (7-16)	Name Type (17)	Reserved (18)	Name (19-28)	Reference (29)	Length (30-34)	Data Type (35)	Decimal Positions (36-37)	Usage (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00101	A		R		REC1								GETRETAIN
	A												UNLOCK

HELP[(response-indicator ['text'])]

Use this file or record level keyword to specify that you want to recognize and handle the Help key through your program rather than have CPF issue an error message to the user indicating that the Help key is invalid at that time.

The Help key functions as a command attention key (no data is received from the display).

Control does not return to your program if either of the following is true:

- An error message is displayed.
- A message subfile is displayed and the cursor is positioned in it.

Note: On the system console, if the cursor is positioned within the error message, the Help key displays the second-level text of the system message, not the second-level text of the program-generated message.

The optional text is included on the listing generated at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The apostrophes are required. If you specify more than 50 characters between the apostrophes, the text is truncated to 50 characters in the program listing.

One use of this keyword is to specify the same indicator as the response indicator for the HELP keyword and as an option indicator on a message field. When the user presses the Help key, the indicator is turned on and control returns to the program. The program can then issue another put operation to the record format on the screen; with the indicator on, the message is displayed on the message line.

A second use of this keyword is to save any keyed input and display help text on a new display.

Option indicators are valid with this keyword.

The following example shows how to specify the HELP keyword:

A

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00024	A																		HELP(99 'Help Key')
	A																		
	A																		

HOME[(response-indicator ['text'])]

Use this file or record level keyword to specify that you want to recognize and handle the Home key through your program.

If the work station user presses the Home key and the cursor is not already at the home position, the cursor returns to the home position, whether the HOME keyword is specified or not.

If the cursor is already at the home position when the Home key is pressed, CPF returns control to your program as it does when a command attention key is pressed (no data is received from the device). In this situation, if you have not specified the HOME keyword, CPF issues a message indicating that the key is invalid at that time.

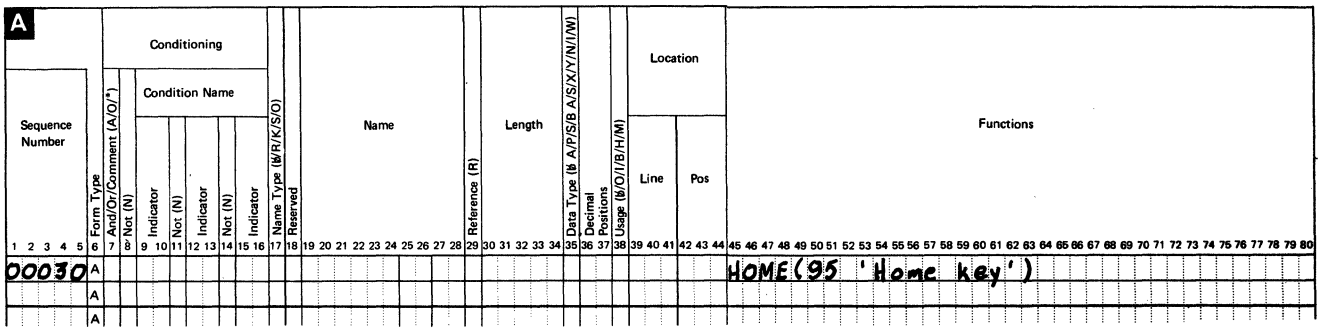
The home position is one of the following (in order of priority):

- The cursor position specified by the last output operation
- The first unprotected input field
- Position 1, line 1

The optional text is included on the listing generated at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The apostrophes are required. If you specify more than 50 characters between the apostrophes, the text is truncated to 50 characters on the program listing.

Option indicators are valid with this keyword.

The following example shows how to specify the HOME keyword so that if the cursor is in the home position, when the Home key is pressed control returns to the program with response indicator 95 set on:



INDTXT(indicator 'indicator-usage-text')

Use this file, record and field level keyword to associate descriptive text (indicating intent or usage) to a specific response or option indicator. You can specify it once for each response and option indicator.

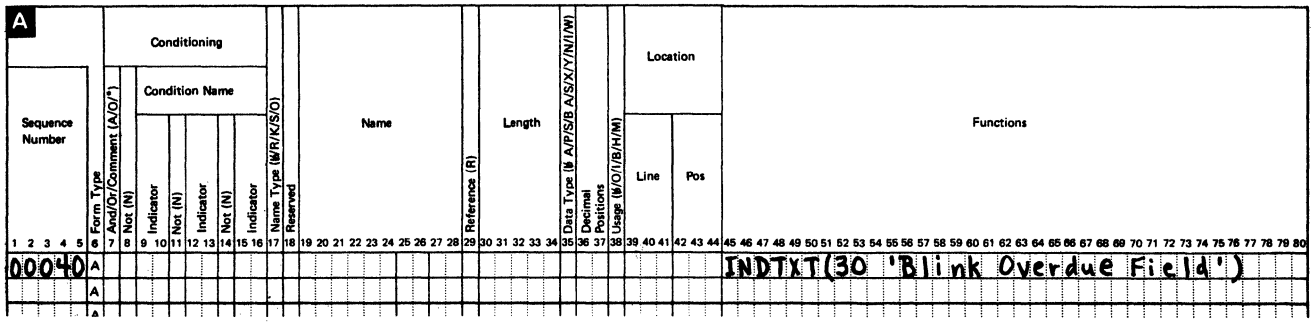
If you specify this keyword, indicator usage text is a required parameter value and must be a character literal enclosed in apostrophes.

The text is included on the listing generated at program compilation time to explain the intended use of the indicator. This text has no function in the file nor the program other than as a comment. The apostrophes are required. If you specify more than 50 characters between the apostrophes, the text is truncated to 50 characters on the program listing.

Option indicators are not valid with this keyword.

Note: The INDTXT keyword does not by itself cause the specified indicator to appear in either the input or the output record area. It merely provides text to be associated with the indicator. If the indicator has not been specified elsewhere (as either an option indicator or a response indicator), then the text is lost without a diagnostic. Also, once an indicator has been given a text assignment (either by this keyword or the response indicator text), no other text assignment is allowed.

The following example shows how to specify the INDTXT keyword:



INZRCD

Use this record level keyword (initialize record) to specify that if this record is not already on the screen, it is to be written to the screen before an input operation is issued from the program specifying this record name. If this record is already on the screen, the keyword is ignored. This implicit output operation, when performed, is CPF-initiated; its purpose is only to format the screen when an input operation is performed. Option indicators are not valid.

This keyword does not apply to output operations.

The following example shows how to specify the INZRCD keyword:

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9–10)	Not (N) (11)	Indicator (12–13)	Not (N) (14)	Indicator (15–16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19–28)	Reference (R) (29)	Length (30–34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36–37)	Usage (b/O/I/B/H/M) (38)	Location Line (39–41)	Location Pos (42–44)	Functions (45–80)
00274	A								R		REC4								INZRCD
	A																		
	A																		

If INZRCD is not specified, your program is given an error if it tries to read a record when that record is not on the screen.

When the INZRCD keyword is processed, the following special conditions exist:

- For output-only fields, no user data is available. The field appears on the screen as blanks. Any editing specified is ignored. The BLKFOLD keyword does not affect the screen.
- For output/input fields, no user data is available. The field appears on the screen as blanks. Any editing specified is ignored. The input save area is initialized in the same way as uninitialized input-only fields (blanks or zeros, depending on the data type).
- Constants and input-only fields appear the same as when displayed using an explicit output operation.
- Hidden fields are returned on a input operation as blanks or zeros.
- Message fields are ignored; there is no message data.
- The LOGOUT keyword is ignored because there is no output buffer to log.
- The ERRMSG and ERRMSGID keywords are ignored because the record is not already on the screen.
- The SFLMSG and SFLMSGID keywords are ignored.
- All other optioned keywords and fields are processed as if they were optioned.

Note: Your program must specify a record format name when issuing an input operation that contains this keyword for the INZRCD function to be performed.

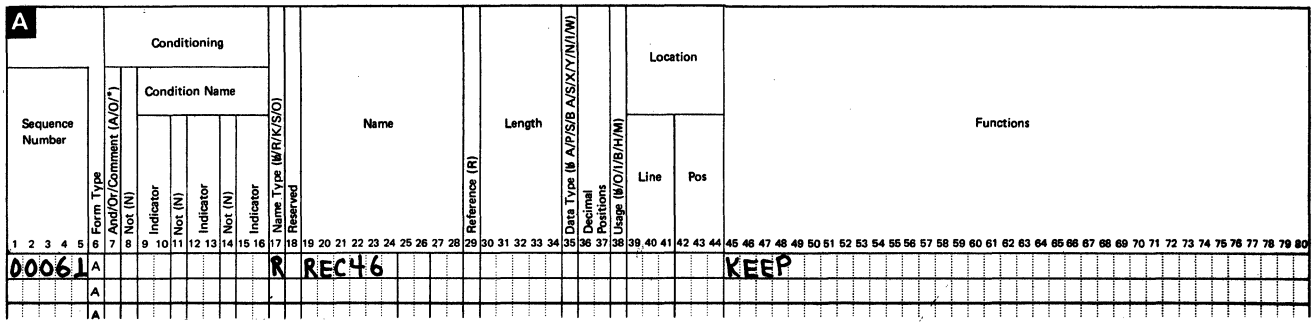
KEEP

Use this record level keyword to keep the screen from being erased when the display file is closed. The entire screen is kept if any of the records on the screen have the KEEP keyword specified; the default causes the entire screen to be erased when the file is closed.

In addition, the name of the first, uppermost record on the screen that has the KEEP attribute is saved by CPF for possible use by subsequent programs. The name that is kept can be used by a subsequent program that does not specify a record name on its first input operation. This keyword enables you to leave data on the screen for review after your program terminates, or you can use that data as input for subsequent programs. For additional considerations for this keyword, see the *CPF Programmer's Guide*.

Option and response indicators are not valid with this keyword.

The following example shows how to specify the KEEP keyword:



This record level keyword is used to specify that the keyboard is to remain locked after an output operation. Normally, the keyboard is automatically unlocked after an output operation.

The LOCK keyword is used when there are several consecutive output operations that contain input fields. If this keyword is not specified, the work station user could key data into a field when a subsequent output operation sends data to the screen. In this case, the cursor location can be changed and the key entry data lost.

If one program performs an output operation that unlocks the keyboard and a subsequent program attempts to open a different display file, the request to open that file is not accepted by data management. (See the *CPF Programmer's Guide* for additional considerations.)

Note: The default on an output operation is to unlock the keyboard. If the keyboard is locked when an input operation is issued, it is automatically unlocked.

This keyword is independent of other keywords that affect the output operation.

Option indicators are valid with this keyword.

The following example shows how to specify the LOCK keyword:

A

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00011	A								R		REC1								LOCK
	A																		
	A																		
	A																		
	A																		

LOGINP

Use this record level keyword to specify that the input buffer for this record format is to be written to the job log each time CPF performs an input operation for the record. The data logged includes the values of input-capable fields, hidden fields, and response indicators specified in the record format you are defining. The function of this keyword is for debugging and other exception conditions. The job log cannot be read by your program.

CPF ignores LOGINP if there are no input-capable or hidden fields or if there are no response indicators in the record format.

Option indicators are not valid with this keyword.

The following example shows how to specify the LOGINP keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/(N/I/N))	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00072	A								R		REC24								LOGINP
00073	A																		
	A																		
	A																		
	A																		

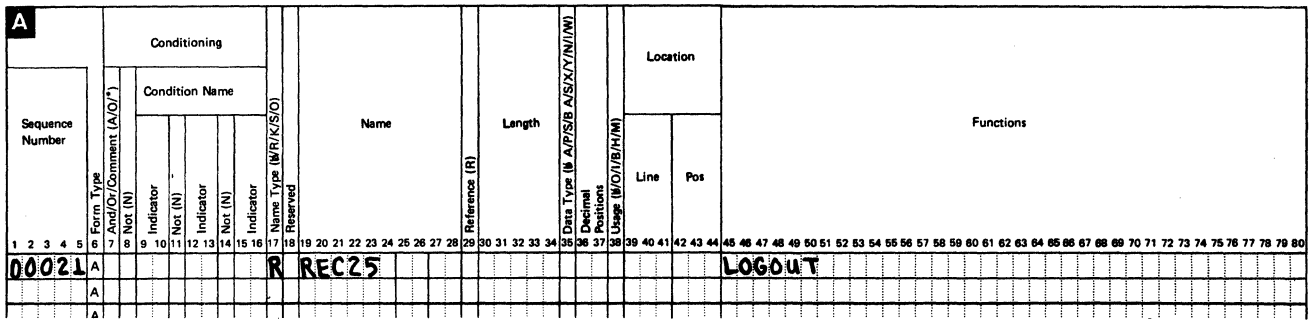
LOGOUT

Use this record level keyword to specify that the output buffer for this record format is to be written to the job log each time CPF performs an output operation for the record. The data logged includes the values of output-capable fields, hidden fields, and option indicators specified in the record format you are defining. The function of the LOGOUT keyword is for debugging and other exception conditions. The job log cannot be read by your program.

CPF ignores the LOGOUT keyword if there are no output-capable or hidden fields, and if there are no option indicators in the record format.

Option indicators are valid with this keyword.

The following example shows how to specify the LOGOUT keyword:



LOWER

The LOWER keyword is equivalent to the CHECK(LC) keyword. The CHECK keyword is preferred because it is compatible with DDS on other systems. See *Keyboard Control* in the CHECK keyword description for an explanation of how to use this keyword.

MDTOFF [{ *UNPR / *ALL }]

Use this record level keyword with the OVERLAY keyword to set off modified data tags (MDTs) for input-capable fields in record formats already on the screen. The MDTs are set off when your program issues an output operation to the record format you are defining. To set off MDTs for unprotected fields only (those without DSPATR(PR) in effect), specify the *UNPR parameter value (this is also the default if no parameter value is specified). To set off MDTs for all input-capable fields, specify the *ALL parameter value.

Your program can select DSPATR(MDT) for fields in the same record format for which it selects MDTOFF (any parameter). If so, these fields are displayed with their MDTs set on.

On the system console, the MDTOFF(*UNPR) keyword is treated as the MDTOFF(*ALL) keyword.

ERASEINP(*ALL) implies MDTOFF(*UNPR) unless MDTOFF(*ALL) is specified.

If ERASEINP(*MDTON) is specified with MDTOFF(*ALL), the end effect is as if ERASEINP(*ALL) and MDTOFF(*ALL) are both specified. This is also true if ERASEINP is specified with no parameter value.

Option indicators are valid with this keyword.

The MDTOFF keyword is not valid for the subfile record format (identified by the SFL keyword); it is valid for all other record formats for which OVERLAY is also specified.

The following example shows how to specify MDTOFF keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		RECORD1								OVERLAY MDTOFF
00020	A										FLD1		6			B	2	2	
00030	A										FLD2		6			B	3	2	
00040	A	*																	
00050	A								R		RECORD2								OVERLAY MDTOFF(*UNPR)
00060	A										FLD21		6			B	4	2	
00070	A										FLD22		6			B	5	2	
00080	A	*																	
00090	A								R		RECORD3								OVERLAY MDTOFF(*ALL)
00100	A										FLD31		6			B	6	2	
00110	A										FLD32		6			B	7	2	
00120	A										FLD33		6			B	8	2	DSPATR(PR)
	A																		
	A																		

In this example, RECORD1 and RECORD2 have equivalent MDTOFF specifications. When RECORD1 or RECORD2 is displayed, the MDT of each input-capable field already on the screen is set off, unless the field has the DSPATR(PR) keyword in effect (FLD33, when displayed, is such a field).

When RECORD3 is displayed, the MDTs of each input-capable field already on the screen is set off even if DSPATR(PR) is in effect for the field.

MSGLOC(line-number)

Use this file level keyword to move the message line to the specified line number. If MSGLOC is not specified, the message line is the last line of the display screen. The message line is the display location for the following messages:

- Validity check errors
- Invalid function keys
- Messages defined as parameter values for the ERRMSG and SFLMSG keywords
- First-level text for messages identified by the ERRMSGID and SFLMSGID keywords (second-level text requires the entire screen)
- Message fields
- Operator error codes and their associated messages (5250 work stations only)

The MSGLOC specification is in effect continuously from file open to file close. It can be temporarily overridden if the file you are defining is suspended while another file is opened to the same work station device. The message location in effect for the other file is used until the file you are defining is restored. See the section on using record formats in different display files in the *CPF Programmer's Guide*.

The parameter value is required and can be any one- or two-digit number in the range of 1 through 24. The number must not be greater than the maximum line number for the screen size being used. Screen size condition names must be specified if message lines for secondary screen sizes are to be different from the last line of the display screen.

Option indicators are not valid with this keyword.

On the 5250 work stations, any data on the message line before the message appears is saved and restored after the Reset key is pressed.

On the system console, any data displayed on the message line before the message appears is erased when the message appears. Pressing the Reset key does not restore the original data. It is recommended that fields not be defined to appear on the message line for the system console.

The following example shows how to specify the MSGLOC keyword:

A

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Conditioning: Not (N) / Indicator / Condition Name (8–16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19–28)	Reference (R) (29)	Length (30–34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36–37)	Usage (b/O/I/B/H/M) (38)	Location Line (39–41)	Location Pos (42–44)	Functions (45–80)
00010	A		*DS2											MSGLOC(10)
	A													
	A													
	A													
	A													

OPENPRT

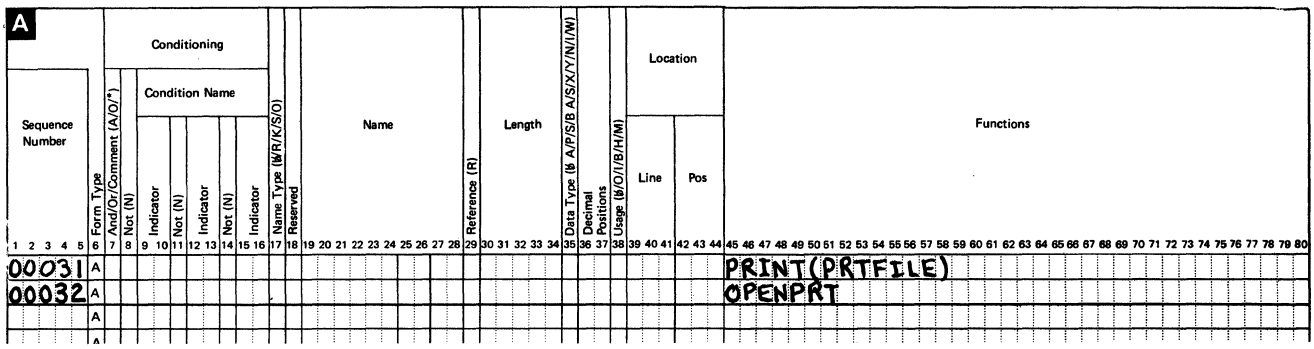
Use this file level keyword to specify that once the printer file is opened (first time the user presses the Print key [on the 5250 work station] or the CF24 key [on the system console]), it is to remain open until the associated display file is closed. If you do not specify OPENPRT (and the PRINT keyword is specified), the printer file is opened and closed each time a screen image is printed.

The printer file should be spooled if more than one job uses the same printer file and device. While the printer file is open in the nonspooled mode, the associated printer is allocated to the program or process using this function.

This keyword is valid only if you have specified the PRINT keyword.

Option indicators are not valid with this keyword.

The following example shows how to specify the OPENPRT keyword:



OVERLAY

Use this record level keyword to specify that the record format you are defining should appear on the display without the entire display being erased first. Normally, the entire display is erased on each output operation. All records on the display with fields that partially or completely overlap fields in this record are erased before this record is displayed; all others remain on the display and are not modified in any way. (A record already on the display is erased even if fields specified in the record format are not selected for display.) For example, assume that the following records are on the display:

REC1 (lines 1 and 2)
REC2 (lines 3 and 4)
REC3 (line 5)
REC4 (line 9)

An output of REC5 (lines 4 and 5) with OVERLAY would leave the display with the following records:

REC1 (lines 1 and 2)
REC5 (lines 4 and 5)
REC4 (line 9)

If the record with OVERLAY in effect is already on the display and PUTOVR, PUTRETAIN, or CLRL is not specified, it is erased and rewritten as a new record.

When the beginning attribute character of a record overlaps the ending attribute character of a record already displayed, the record already displayed is erased unless the attribute characters overlap each other in position 1 of a line. (The last field of the first record displayed ends in the rightmost display position of the preceding line.)

In the above example, however, if the only portion of REC2 on line 4 is the ending attribute character of the last field of REC2 (which occurs when the last displayed character of the last field of REC2 is in the last position of line 3), REC2 remains displayed following the display of REC5 with OVERLAY. The display would have the following records:

REC1 (lines 1 and 2)
REC2 (line3)
REC5 (lines 4 and 5)
REC9 (line 9)

The display is always erased on the first output operation after the file is opened, except when both ASSUME and OVERLAY are specified.

OVERLAY is assumed by CPF for ERRMSG, ERRMSGID, PUTOVR, and CLRL functions.

To erase any of the records on the display, use the ERASE keyword to specify the names of the record formats to be erased.

If you also specify the CLRL keyword, execution proceeds according to the CLRL specification, not the OVERLAY specification.

If OVERLAY is conditioned and not selected, then ERASE, ERASEINP, MDTOFF, PROTECT, and PUTRETAIN cannot take effect if they are selected, unless PUTOVR is selected. In such cases, ERASE, ERASEINP, and MDTOFF can take effect.

Option indicators are valid with this keyword.

See the *CPF Programmer's Guide* for more information on the use of the OVERLAY keyword.

The following example shows how to specify the OVERLAY keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00021	A								R		RECL								OVERLAY
	A																		
	A																		
	A																		
	A																		
	A																		

OVRATR

Use this field level keyword with the PUTOVR keyword to override the existing display attributes of a field already on the display. The OVRATR keyword can be used with the OVRDTA keyword on the same field.

The display attributes that can be overridden by the OVRATR keyword are:

CHECK(ER)
CHECK(ME)
DSPATR (all except OID and SP)
DUP

The OVRATR keyword is valid only with input-only, output-only, output/input, or constant fields.

When the OVRDTA keyword is in effect, the display attribute can also be overridden on the same output operation (as if the OVRATR keyword were also in effect).

Option indicators are valid with this keyword.

For a discussion and example of how to use the OVRATR keyword, see the PUTOVR keyword description in this chapter.

OVRDTA

Use this field level keyword with the PUTOVR keyword to override the existing data contents of a field already on the display. The OVRDTA keyword can be used with the OVRATR keyword on the same field.

The OVRDTA keyword is valid only with output-only, output/input, or message fields.

The OVRDTA keyword is required if the DFT keyword is specified for output-only or output/input fields.

Option indicators are valid with this keyword.

For a discussion and example of how to use the OVRDTA keyword, see the PUTOVR keyword description in this chapter.

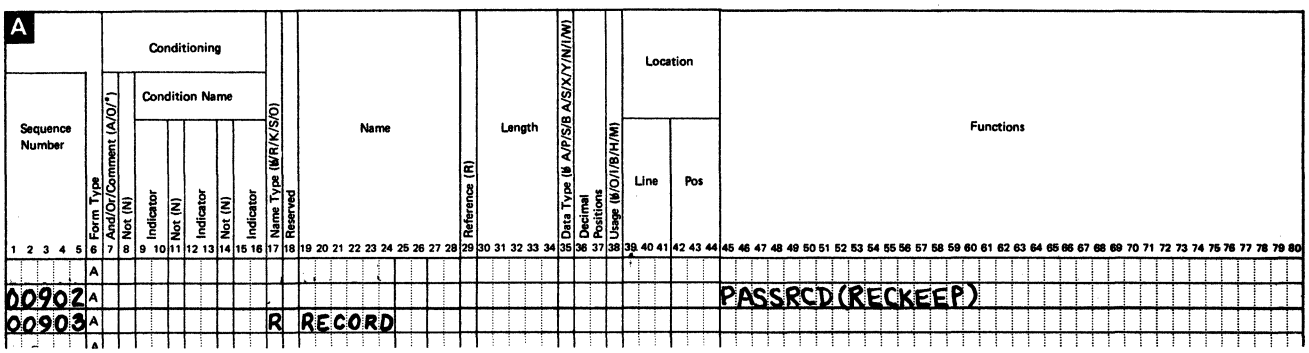
PASSRCD(record-format-name)

Use this file level keyword to specify the record format to be used by CPF when unformatted data is passed to your program by another program. The passed data is processed *only* if the first request from your program after file open is an input operation without a record format name. The data must be processed according to this record format. (See the *CPF Programmer's Guide* for more information.)

The record format name is a required parameter value for this keyword.

Option indicators are not valid with this keyword.

The following example shows how to specify the PASSRCD keyword:



PRINT[(response-indicator ['text']) | (device-file-name[.library-name])]

Use this file level keyword to specify that the work station user can press the Print key (or, on the system console, the CF24 key) to print the current display. The following three examples illustrate the three ways of specifying the PRINT keyword:

PRINT	CPF prints on QSYSPRT. For the 5250 work station, you can specify another printer file on the Create Device Description (CRTDEVD) or Change Device Description (CHGDEVD) commands. See *PRINT Keyword without Parameter Values* later in this keyword description.
PRINT(01 'User presses Print key')	Your program is given control and decides what to do (for example, produce formatted printer output). The response indicator is set on. No data is sent from the device. (On the system console, the key labeled CF24 acts like CA24.)
PRINT(PRINTFILE1.LIB1)	CPF prints the display through the specified printer file (which can be described through DDS). An Override with Printer File (OVRPRTF) command, if in effect before the printer file is opened (when the Print key is pressed), can change the printer device name.

Option indicators are valid with this keyword.

Further considerations regarding the three ways of specifying the PRINT keyword are discussed in the following sections.

PRINT Keyword without Parameter Values

For the Local 5250 Work Station: CPF attempts to print the screen image on the associated work station printer without sending the data through the system. Nondisplay fields appear as blanks; duplicated characters entered by pressing the Dup key appear as asterisks (*); screen attributes appear as blanks. If the print function cannot be performed successfully, CPF attempts to complete the print function, using the printer file specified on the PRTFILE parameter on the CRTDEVD or the CHGDEVD command used to describe the display device to the system.

For work station printers attached through the work station controller, a message indicating that there is a problem is issued to the work station user requesting the print function. The work station user can make the printer ready or press the Reset key. To cancel a print request before it is complete, the work station user can press the shifted Print key.

Note: After the current display is printed, the paper is advanced twice the number of lines as in the current screen size (24 lines for a 12 by 80 screen and 48 lines for a 24 by 80 screen).

For the Remote 5250 Work Station: If the printer is not ready when the Print key is pressed, no specific message is sent to the user. The work station requesting the print function remains inoperable until the printer is made ready, or until the print request is canceled (by using the shifted Print key).

Note: After the current display is printed, the paper is advanced the same number of lines as in the current screen size (12 lines for a 12 by 80 screen and 24 lines for a 24 by 80 screen).

For the System Console: CPF opens QSYSPRT (a spooled output file), reads the screen buffer, and directs the screen image to QSYSPRT. The file is opened for each printer request and is closed when the print function is complete.

The following example shows how to specify the PRINT keyword with no parameter values:

A

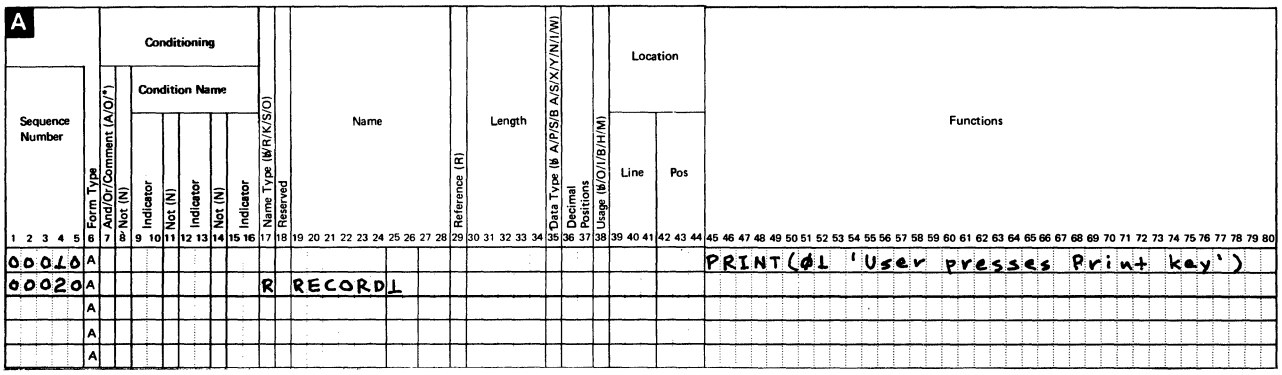
Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00010	A																		PRINT
00020	A								R		RECORD1								
	A																		
	A																		
	A																		

PRINT Keyword with Response Indicator

If you specify the PRINT keyword with a response indicator, CPF returns control to your program with the specified response indicator set on. No data is received from the device. The keyboard on the 5250 work station is locked until your program issues another output operation to the display file. The keyboard on the system console is not locked. There is no difference in the print function between local and remote 5250 work stations.

The optional text is included on the listing generated at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The apostrophes are required. If you specify more than 50 characters between the apostrophes, the text is truncated to 50 characters on the program listing.

The following example shows how to specify the PRINT keyword with a response indicator:



A	Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
	1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	00010	A																		PRINT(01 'User presses Print key')
	00020	A								R		RECORD1								
		A																		
		A																		
		A																		

PRINT Keyword with a Printer File Specified

CPF reads the screen buffer and prints the screen image using the specified printer file. The printer file that you specify as a parameter value for this keyword can be either an externally described or a program-described file. It also can be either spooled or nonspooled. If you specify an externally described printer file, it must contain a record format with the same name as the file.

The printer file must exist and be authorized to the user of the display when the Print key is pressed. This also applies to the library name if it is specified. If CPF is unable to perform the print function on the specified printer file, CPF attempts to use the printer file specified on the PRTFILE parameter of the CRTDEVD or the CHGDEVD command. It is recommended that SPOOL(*YES) be specified on the CRTDEVD or the CHGDEVD command to prevent locking of the 5250 work station keyboard (the keyboard on the system console is not locked).

If you do not specify the library name, the current library list is used.

See the OPENPRT keyword for details of when the printer device file is opened and closed.

The following example shows how to specify that the display is to be directed to printer file PRINTFILE1.LIB1:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/N)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A																		PRINT(PRINTFILE1.LIB1)
00020	A								R		RECORD1								
	A																		
	A																		
	A																		

PROTECT

Use this record level keyword with the OVERLAY keyword to specify that when the record you are defining is displayed, all input-capable fields already on the display are to be changed to output-only fields. This protects them from input keying. The data contents of the input-capable fields are not changed, but your program cannot read them unless it first redisplays the record formats in which the input-capable fields are specified.

To protect a single field from input keying, see the DSPATR(PR) keyword.

The PROTECT keyword is valid only for the 5250 work station and is ignored for the system console.

The OVERLAY keyword must be specified in the record format in which PROTECT is specified. Also, either the OVERLAY keyword or the CLRL keyword must be in effect for the PROTECT keyword to be in effect.

PROTECT is not in effect if PUTOVR is in effect.

If the ERASEINP and PROTECT keywords are both in effect for an output operation, CPF first erases the input-capable fields specified on the ERASEINP parameter value, then protects all input-capable fields on the display from input keying.

Option indicators are valid with the PROTECT keyword.

The following example shows how to specify the PROTECT keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B/A/S/X/Y/N/I/N/I/N)	Decimal Positions	Usage (blank/O/I/I/B/I/H/M)	Location: Line	Location: Pos	Functions
00010	A								R		RECORD1								
00020	A										FLD1		5			I	5	3	
00030	A																		
00040	A								R		RECORD2								
00050	A																		OVERLAY
00060	A			32							FLDA		10			I	6	3	PROTECT
	A																		
	A																		
	A																		

In this example, RECORD1 has an input-capable field that has been displayed and read and that should be left on the display while RECORD2 is displayed and read. To prevent further keying into FLD1 in RECORD1, issue an output operation to RECORD2 a second time with the PROTECT keyword in effect. Note that FLDA is not protected, but FLD1 is.

PUTOVR

Use this record level keyword (put with explicit override) to permit the override of either display attributes or data contents (or both) of selected fields within a record displayed on a work station device. By using the PUTOVR keyword, you can reduce the amount of data sent to the display device.

When selected fields in a record that has already been displayed are to be changed, an output or an output/input operation issued to the record with PUTOVR, OVRDTA, and OVRATR in effect changes only the fields for which OVRDTA or OVRATR is in effect. The OVRDTA keyword permits a change in the data contents of the field and the OVRATR keyword permits a change in the display attributes of the field. The way in which fields are to be changed is controlled by setting option indicators.

Option indicators are valid with the PUTOVR, OVRATR, and OVRDTA keywords.

The following conditions cause the put-override keywords to be ignored and no error to occur:

- PUTOVR is not in effect at the time of the output operation.
- Both OVRDTA and OVRATR are not in effect at the time of the output operation.
- The record is not already on the screen.

The PUTOVR and OVRDTA keywords must be specified when DFT is specified for a named output-capable field. When PUTOVR and OVRDTA are both in effect for a field, the default value specified with the DFT keyword is displayed only on the first display of the field. On subsequent displays with PUTOVR and OVRDTA in effect, the program value is displayed.

CPF assumes that, except for input-capable fields, all fields specified to be overridden are already displayed. This assumption differs from PUTRETAIN.

The PUTRETAIN keyword and the put-override keywords are mutually exclusive within the same record format.

The OVRDTA keyword is permitted only with output-only, input-only, output/input, or message fields (usage O, I, B, or M, respectively).

The OVRATR keyword is permitted only with output-only, input-only, or output/input fields (usage O, I, or B, respectively).

The OVRATR keyword can be used only to override the following display attributes:

CHECK(ER)
CHECK(ME)
DUP
DSPATR (all except OID and SP)

An output operation with OVRDTA in effect does not need to have OVRATR in effect to override display attributes, as well as data contents, of the field or fields being overridden.

For example, if a work station user has keyed data into input-capable fields but has not yet pressed the Enter/Rec Adv key, the program can change the display attributes or displayed data within certain fields of the display by issuing an output operation with the appropriate put-override keywords selected.

A

```
00010A                 R INVRCD1
00020A                                            PUTOVR
00030A                                          1 26'INVENTORY REMAINING IN WAREHOUSE 1'
00040A*
00050A                                          3  1'Remaining on hand'
00060A   11                                       OVRATR DSPATR(HI)
00070A*
00080A                   INVBAL         5Y 0      +3
00090A   12                                       OVRDTA
00100A*
00110A                   SUPPLIER      20   B   5  1DFT('INTERNAL')
00120A   13                                       OVRDTA
00130A*
00140A                   ACCOUNT       20       6  1DSPATR(HI)
00150A                                            OVRDTA
00160A   14                                       DSPATR(RI)
```

In this example, an initial output operation displays a full screen of information; on a second output operation, PUTOVR is in effect and the program can set option indicators to make the following changes to the display:

- If option indicator 11 is set on, the constant 'Remaining on hand:' would be changed to a highlighted field. (The only way to reset the display attribute to normal is to redisplay the record format after erasing it.)
- If option indicator 12 is set on, the program can change the displayed value of the field INVBAL.
- If option indicator 13 is set on, the program can set the value of the field SUPPLIER to override the default value ('INTERNAL').
- If option indicator 14 is set on, the display attribute of the field ACCOUNT is changed from highlight to highlight and reverse image at the same time that new data is sent to the field. If option indicator 14 is set off, the display attribute is changed back to highlight. New data is sent to the display on each output operation.

PUTRETAIN

Use this record or field level keyword with the OVERLAY keyword so that CPF does not erase data already on the screen when redisplaying a record. The PUTOVR keyword has a function similar to, but more effective than, this keyword.

To understand what effect this keyword has on output operations, consider the following sequence of steps:

1. Your program issues an output operation to RECORD1, displaying RECORD1. The PUTRETAIN keyword, if in effect, is ignored. Any data in the record area for RECORD1 is erased before RECORD1 is displayed.

2. At some later time, with RECORD1 still on the screen, your program issues a second output operation to RECORD1. Two conditions can occur:

   - If the PUTRETAIN keyword is not in effect, CPF first erases the record area for RECORD1, then displays RECORD1. Fields selected for display at this time are displayed with new data contents and new display attributes, which can be the same as before. The record area includes every line on which a field or part of a field for RECORD1 appears.

   - If the PUTRETAIN keyword is in effect, CPF does not erase the record area for RECORD1. The data contents of selected fields are not changed. However, the display attributes for selected fields are sent to the screen and can be changed (by selecting which DSPATR keyword is in effect for this output operation). Fields not selected for display are written over character by character by fields selected for display. For more information, see *When Fields are Selected by Option Indicators* later in this keyword description.

Option indicators are valid with the PUTRETAIN keyword.

Conditions Affecting the PUTRETAIN Keyword: The PUTRETAIN keyword applies only to the record format for which it is specified, and only if the record is already displayed on the screen. If the record on which the PUTRETAIN keyword is specified is not on the screen, the PUTRETAIN keyword is ignored.

When specified at the record level, this keyword applies to all fields in the record format that have been selected for display. When specified at the field level, this keyword applies only to the fields for which it is specified.

This keyword can be specified for more than one field of a record format, but only once per field. This keyword can be specified at the record level and at the field level within the same record format.

The PUTRETAIN keyword cannot be specified with the PUTOVR keyword.

The OVERLAY keyword must be specified when the PUTRETAIN keyword is specified.

If the OVERLAY keyword is not in effect, the PUTRETAIN keyword is ignored and the entire screen erased before the record is displayed.

When Fields are Selected by Option Indicators: When the PUTRETAIN keyword is in effect on an output operation involving field selection, fields in the record format that are not selected for redisplay are not erased from the screen; they can be partially or completely rewritten by newly selected fields.

If PUTRETAIN is in effect only for a newly selected field (specified at the field level), only the beginning attribute character of the field is sent to the screen; the ending attribute character is not sent to the screen. (For fields without PUTRETAIN in the same record format, CPF sends the display attribute and the data.) If the PUTRETAIN keyword is in effect for the whole record (specified at the record level), only the beginning and ending attribute characters are sent to the screen. Thus, the display attribute of a field can be reset to normal if the field immediately preceding this field is selected and this field is not selected. For example, assume that DSPATR(UL) is in effect for two consecutive fields with overlapping attribute characters. If on an output operation with PUTRETAIN in effect, the first of these fields is selected and the second field is not selected, the display attribute of the second field is reset to the normal display attribute. This is because CPF sends the first field to the screen with beginning and ending attribute characters, and its ending attribute character overrides the beginning attribute character of the second field.

The following example shows how to specify the PUTRETAIN keyword (for additional examples of the PUTRETAIN keyword at the record and field level, see the *CPF Programmer's Guide*):

Sequence Number	Form Type	Name Type	Name	Functions
00101	A	R	CUST	
00102	A			PUTRETAIN OVERLAY

RANGE(low high)

Use this field level keyword for input-capable fields to specify that CPF is to perform validity checking on the data that the work station user keys into the field. The data keyed must be greater than or equal to the lower value, and less than or equal to the higher value.

When the field is a character field, the parameter values must be enclosed in apostrophes. When the field is numeric, apostrophes must not be specified.

Option indicators are not valid with this keyword.

The following example shows how to specify the RANGE keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A								R		RECORD1								
00020	A	X		Character fields															
00030	A										FIELD1		1			I	2		2RANGE('B' 'F')
00040	A										FIELD2		1			I	3		2RANGE('2' '5')
00050	A	X		Numeric fields															
00070	A										FIELD3		1			ØI	4		2RANGE(2 5)
00080	A										FIELD4		4			ØB	5		2RANGE(1 1500)
00090	A										FIELD5		7			2B	6		2RANGE(100 99999.99)
00100	A										FIELD6		3			ØB	7		2RANGE(-100 -50)
00110	A										FIELD7		3			2I	8		2RANGE(.50 1.00)
00120	A										FIELD8		3			2I	9		2RANGE(.5 1)
00130	A										FIELD9		5Y			2I	10		2RANGE(.01 999.99)
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

In this example, FIELD7 and FIELD8 have equivalent RANGE parameter values. The reason is that for numeric fields, decimal alignment is based on the number of decimal positions specified in positions 36 through 37. For FIELD7 and FIELD8, the low value is 0.50 and the high value is 1.00.

Data entered into a numeric field is aligned on the decimal positions specified (in positions 36 through 37), and leading and trailing blanks are filled with zeros. For example, if 1.2 is keyed into FIELD9, 00120 is returned to your program. If 100 is keyed into FIELD9, 10000 is returned to your program.

REF(data-base-file-name[.library-name][record-format-name])

Use this file level keyword to specify the name of a file from which field descriptions are to be retrieved. Use it when you want to duplicate descriptive information from several fields in a previously described record format. You can code the file name once here rather than on REFFLD keywords with each of the field descriptions that reference the file. To reference more than one file, use the REFFLD keyword. (The REF keyword can be specified only once.)

If there is more than one record format in the referenced file, specify a record format name as a parameter value for this keyword to tell CPF which one to use unless the record formats should be searched sequentially.

The file name is a required parameter value for this keyword; the library name and the record format name are optional.

If you do not specify the library name, the current library list is used. If the record format name is not specified, each record format is searched in order (as they are specified); the first occurrence of the field name is used. See Appendix A, *How to Specify REF and REFFLD,* for the search sequences determined by your choice of REF and REFFLD keywords.

Option indicators are not valid with this keyword.

The following examples show how to specify the REF keyword:

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (b/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (b A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (b/O/I/B/H/M)	Location Line	Location Pos	Functions
00010	A																		REF(FILE1)
00020	A								R		RECORD								
00030	A										FLD1	R					2	2	
	A																		
	A																		
	A																		

In this example, FLD1 has the same attributes as the first (or only) FLD1 in FILE1.

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (b/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (b A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (b/O/I/B/H/M)	Location Line	Location Pos	Functions
00010	A																		REF(FILE1.LIB1 RECORD2)
00020	A								R		RECORD								
00030	A										FLD1	R					2	2	
	A																		
	A																		
	A																		

In this example, FLD1 has the same attributes as FLD1 in RECORD2 in FILE1 in LIB1.

REFFLD(referenced-field-name[.record-format-name] {*SRC / data-base-file name[.library-name]})

Use this field level keyword when referencing a field under one of these three conditions:

- The name of the referenced field is different from the name in positions 19 through 28.
- The name of the referenced field is the same as the name in positions 19 through 28, but the record format, file, or library of the referenced field is different from that specified with the REF keyword.
- The referenced field occurs in the same DDS source file as the referencing field.

The referenced field name is required even if it is the same as the referencing field. Use the record format name when the referenced file contains more than one record format. Use *SRC to search the DDS source file in which the referencing field occurs (this is the default value when the data base file name and library name are not specified). Specify the data base file name (qualified by its library name if necessary) when you want to search a particular data base file.

If, in the same DDS source file, you specify REF at the file level and REFFLD at the field level, the REFFLD specification is used. The particular search sequence depends on both the REF and REFFLD keywords. For more information, see Appendix A, *How to Specify REF and REFFLD*.

An R must be specified in position 29.

Option indicators are not valid with this keyword.

The following example shows how to specify the REFFLD keyword. See the example in Appendix A, *How to Specify REF and REFFLD*, for explanations of the various specifications.

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		RECORD1								
00020	A										ITEM		5					1	
00030	A										ITEM1	R						2	REFFLD(ITEM)
00040	A										ITEM2	R						1 2	REFFLD(ITEM.FMAT1)
00050	A										ITEM3	R						2 2	REFFLD(ITEM FILEX)
00060	A										ITEM4	R						3 2	REFFLD(ITEM FILEX.LIBY)
00070	A										ITEM5	R						4 2	REFFLD(ITEM.FMAT1 FILEX.LIBY)
00080	A										ITEM6	R						5 2	REFFLD(ITEM *SRC)
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

ROLLUP/ROLLDOWN[(response-indicator ['text'])]

Use these file or record level keywords to specify that you want to use your program to handle any situation where the work station user has pressed the specified key and CPF cannot roll the display. If this situation occurs and you have not specified this keyword (whichever keyword is appropriate), CPF issues an error message indicating that the key is not valid at that time.

You can specify a response indicator with this keyword. If you do, and the appropriate roll key is pressed, CPF sets on the specified response indicator within the input record and returns control to your program after it processes the input data. If you do not specify a response indicator and the specified roll key is pressed, CPF performs normal input record processing.

These keys cause data to be returned from the display device to your program (similar to CF and Enter/Rec Adv keys).

If CPF is performing the roll function for subfiles (SFLSIZ value does not equal SFLPAG value), you do not need to specify these keywords. For a description of what happens when ROLLUP and ROLLDOWN are specified for a subfile, see the extended description of the SFLROLVAL keyword.

For additional information on how to use this function, see the *CPF Programmer's Guide.*

The optional text is included on the listing generated at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The apostrophes are required. If you specify more than 50 characters between the apostrophes, the text is truncated to 50 characters on the program listing.

Option indicators are valid with these keywords.

The following example shows how to specify the ROLLDOWN keyword:

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00010	A		N	64															ROLLDOWN(52 'Roll Down')
	A																		
	A																		

Use of Routing Keywords

You can use the routing keywords (RTG prefix) to cause CPF to place specific information regarding input operations (up to 80 characters) in the routing identification field in the I/O feedback area. No information is entered for an output-only operation.

CPF places the information in the routing field only after validity checking, data alignment, and other specified functions have been performed successfully. Option indicators are not valid for routing keywords. Routing keywords are not valid for subfile records. For further information see the appropriate high-level language manual.

Note: Other portions of the feedback area contain information such as device type, cursor address, and the actual AID byte (one character) representing the attention or function key entered (see the RTGAID keyword).

Each routing keyword requires that you specify a *routing field location*, which is the first character position for that item in the routing field within the I/O feedback area. All entries must fit completely within the 80-byte routing area, and no checking is done to prevent overlapping entries in the routing area. If entries do overlap, the results can be predicted, based on the processing order of the keywords. Entries processed later overlay entries processed earlier.

Note: Multiple specifications are not checked for overlap in the I/O feedback area.

The processing order of the routing keywords follows:

RTGCON
RTGDEVCLS
RTGDEV
RTGAID
RTGFMT
RTGFIRST
RTGPOS (only valid when USRDFN also specified)
RTGFLD

Figure 4-8 shows how to specify some of the routing keywords:

```
00010A*
00020A*  USE OF ROUTING KEYWORDS
00030A          R RTGREC                    RTGFMT(33)
00040A                                      RTGAID(1)
00050A                                      RTGCON(3 'ABC')
00060A                                      RTGDEV(10)
00070A                                      RTGDEVCLS(20)
00080A                                      RTGFIRST(22 8)
00090A            CODE           3   I  6  9RTGFLD(30)
```

Figure 4-8. Routing Keyword Coding Example

RTGAID(starting-routing-field-location)

Use this record level keyword to cause CPF to indicate the AID (attention identifier) byte (the command key definition as received from the display) as a 2-byte identifier placed into the routing field as follows:

Key	Identifier
CA or CF 01-24	01-24
Enter Key	RA
Record Advance	RA
Roll Up	UP
Roll Down	DN
Print	PT
Clear	CL
Help	HP
Record Backspace (Home)	BS
Light Pen Auto Enter	LP

Valid starting locations you can specify are within the range 1 through 79.

Figure 4-8 shows how to specify the RTGAID keyword.

RTGCON(starting-routing-field-location 'constant')

Use this record level keyword to specify that CPF is to place the constant that you included as a parameter value with this keyword into the specified locations of the routing area.

Valid locations you can specify are within the range 1 through 80, depending on the length of the literal.

Figure 4-8 shows how to specify the RTGCON keyword.

RTGDEV(starting-routing-field-location)

Use this record level keyword to cause CPF to place the 10-character device name into the routing field. Valid locations you can specify are within the range of 1 through 71.

Figure 4-8 shows how to specify the RTGDEV keyword.

RTGDEVCLS(starting-routing-field-location)

Use this record level keyword to cause CPF to place the 2-character device class indicator into the routing field. Valid locations you can specify are within the range of 1 through 79.

The class values that are placed in the routing field are as follows:

01 System console, 16x64
02 Work station, 12x80
03 Work station, 24x80

Figure 4-8 shows how to specify the RTGDEVCLS keyword.

RTGFIRST(starting-routing-field-location entry-length)

Use this record level keyword to specify that CPF is to take the first field that it receives from the display and place it in the routing area at the specified location using the number of positions that you specified as the entry-length parameter value. If the length of the first field received does not match the field length you specified, CPF either pads or truncates to fit; the pad characters are hex F0 for numeric fields and hex 40 for character fields. If no fields are received from the display, the specified part of the routing area is cleared to blanks (hex 40).

The valid locations you can specify are within the range 1 through 80, depending on the entry length.

Note: Only the first field received from the display can satisfy this keyword. In addition, the location of the first field (line and position) must match the location of a field described in this record. If the location of the first field does not match a field of this record, the RTG field is updated as if no fields were received from the display.

Figure 4-8 shows how to specify the RTGFIRST keyword.

RTGFLD(starting-routing-field-location)

Use this field level keyword (routing field) to specify that the field you are defining is to be a routing field and the contents of the field are to be placed in the routing area, starting at the location you specified with this keyword.

You can specify the RTGFLD keyword only once for each field description. It is valid only for input-capable fields. The contents of this field are placed in the routing-field routing area whether or not the field was selected on the output operation, and whether or not the user entered any data into the field. If the field contains no data when read, the area is initialized with hexadecimal F0s for numeric fields and with hexadecimal 40s for character fields.

The valid locations you can specify with this keyword are in the range of 1 through 80, depending on the field length.

Figure 4-8 shows how to specify the RTGFLD keyword.

RTGFMT(starting-routing-field-location)

Use this record level keyword (routing format) to cause CPF to place the record format name (as determined by CPF for the input operation) into the routing field. The length of this entry in the routing field is always 10 bytes.

Valid field locations you can specify are within the range of 1 through 71.

Figure 4-8 shows how to specify the RTGFMT keyword.

RTGPOS(starting-routing-field-location starting-data-position ending-data-position)

Use this record level keyword to identify character positions (within the data received from the display) that are to be placed into the routing field; the RTGPOS keyword is valid only for user-defined data streams (USRDFN keyword). The positions are specified by supplying the starting position (relative to the beginning of the data stream) and ending position.

Valid field locations you can specify are within the range 1 through 80, depending on starting and ending data locations.

Note: The starting data position and the ending data position are not checked to ensure that they are within the limits of the user-defined data stream.

The following example shows how to specify the RTGPOS keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		USRREC								USRDFN
00020	A																		RTGPOS(1 20 40)
	A																		
	A																		
	A																		
	A																		

RTNDTA

Use this record level keyword to specify that, when your program issues an input operation to this record format, CPF is to return the same data that was returned on the previous input operation issued to this record format. The RTNDTA keyword is ignored if the record format has not already been read. When the RTNDTA keyword is in effect, your program can reread data on the display without requiring CPF to actually pass data from the display device to your program.

The RTNDTA keyword is ignored in the following situations:

- On the input portion of an output/input operation (Put-Get operation)
- On an input operation that is preceded by an output operation to the same record format

The RTNDTA keyword has effect only in the following situation:

- On an input operation following an input operation issued to the same record format without an intervening output operation to that record format.

The following are two uses of the RTNDTA keyword:

- To allow a main program to read a record format that is modified by a work station user. The data read tells the main program which subprogram to call. The subprogram issues an input operation to the same record format, with RTNDTA in effect, to read the same data. This procedure can substitute for passing parameters to subprograms. Note that SHARE(*YES) must be specified for both display files.
- To allow an RPG III program to perform file maintenance with less locking of records in the data base. For instance, the program reads a data base record and displays the record at the display device. The work station user reviews the record, makes any required changes, and presses the Enter/Rec Adv key. While the work station user is making changes, the data base record, if locked, is unavailable to other programs. Hence it is recommended to leave the data base record unlocked. However, when the program reads the record from the display and updates the data base record, the data base record overlays the internal representation of the display record in the program; instead of preventing the overlay by using different field names for the display record and the data base record, the program rereads the display file. With RTNDTA specified, the program retrieves the display record again and can then finish updating the data base.

If the UNLOCK keyword is specified, the RTNDTA keyword cannot be specified.

Option indicators are not valid with this keyword.

The following example shows how to specify the RTNDTA keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		RECORD1								RTNDTA
00020	A										FLD1		5			I	2	2	
00030	A										FLD2		5			B	3	2	
	A																		
	A																		
	A																		
	A																		
	A																		

SETOF(response-indicator ['text'])

Use this record level keyword to specify that when an input operation issued to this record format is completed, the specified response indicator is to be set off.

The optional text is included on the listing generated at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The apostrophes are required. If you specify more than 50 characters between the apostrophes, the text is truncated to 50 characters on the program listing.

This keyword can be used to cause an option indicator that is on for an output operation to be returned in the off condition when the next input operation to the record is completed. (If no input operation is performed, the response indicator remains unchanged.) Your program does not have to turn the indicator off.

Option indicators are not valid with this keyword.

The SETOF is equivalent to SETOFF.

Note: Any indicator is valid for this keyword. It does not have to be previously defined as an option or a response indicator. The indicator becomes a response indicator when you specify SETOF.

If the indicator used with the SETOF keyword is also used with another keyword, such as CHANGE, the on/off status of the indicator is controlled by the other keyword.

The following example shows how to specify the SETOF keyword:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		CUSMST								SETOF(63 'On=display MSG2000 +
00020	A																		CONSOLEMSG')
00030	A										QTYORD		3		0I		5	3	
00040	A			63															ERRMSGID(MSG2000 CONSOLEMSG)
	A																		
	A																		
	A																		

In this example, first-level text of MSG2000 is displayed on the message line when the program issues an output operation to CUSMST with indicator 63 set on. On the next input operation to CUSMST, the SETOF keyword sets off indicator 63. (Indicator 63 is used as both an option and a response indicator.) Specifying SETOF in the DDS eliminates the need to set off the indicator in the program and can simplify RPG program logic. See the *Cobol Reference Manual and Programmer's Guide* for information on handling indicators in COBOL programs.

SETOFF(response-indicator['text'])

The SETOFF keyword is equivalent to the SETOF keyword. The SETOF keyword is preferred because it is compatible with DDS on other systems. See the SETOF keyword description for a description of how to use these keywords.

SFL

Use this record level keyword to specify that this record format is to be a subfile record format. This record format (including its related field descriptions) must immediately precede the subfile control record format (identified by the SFLCTL keyword).

At least one displayable field must be specified in the subfile record format (unless the subfile is a message subfile; see the SFLMSGRCD keyword). The locations specified for fields in this record format are the locations on the screen where the first subfile record in any one page of the subfile is displayed. The rest of the page of records is displayed below the first record. The number of records in a page is determined by the parameter value specified for the SFLPAG keyword. For an explanation of how many display lines are occupied by a subfile, see the SFLPAG keyword description.

Displayable fields specified on the subfile control record format can be displayed at the same time as subfile records; however, fields specified in the subfile control record format cannot overlap fields specified in the subfile record format, even if they are specified with option indicators.

The number of subfiles (each having one SFL and one SFLCTL keyword specified) that can be specified in a display file is limited only by the number of record formats permitted in a display file (1024 record formats, or 512 subfiles, maximum). However, only two subfiles can contain active records or be displayed at one time.

For examples of subfiles, see Appendix F, *Examples*.

Option indicators are not valid with the SFL keyword.

In addition to the SFL keyword, the following keywords are also valid on the subfile record format:

- For message subfiles:

  SFLMSGRCD (required at the record level)
  SFLMSGKEY (required at the field level)
  SFLPGMQ

- For all other subfiles (at the record level only):

  CHANGE
  CHECK(AB)
  CHECK(RL)
  CHGINPDFT
  INDTXT
  KEEP
  LOGINP
  LOGOUT
  SETOF
  SETOFF
  SFLNXTCHG
  TEXT

The following otherwise valid keywords are not valid at the field level when specified for the subfile record format:

DATE
ERRMSG
ERRMSGID
TIME

The following field level keywords are valid on the subfile record format when

SFLPAG equals SFLSIZ

or

SFLPAG does not equal SFLSIZ and these keywords are not conditioned.

The keywords are:

AUTO(RA)
CHECK(ER)
CHECK(ME)
DSPATR(PR)
DSPATR(MDT)
DUP

The following example shows how to specify the SFL keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A								R		SFLR								SFL
	A																		
00020	A										(at least one displayable field)								
	A																		
00030	A								R		SFLCTLR								SFLCTL(SFLR)
00040	A																		SFLPAG(17)
00050	A																		SFLSIZ(17)
00060	A																		SFLDSP
	A																		
	A																		
	A																		

SFLCLR

Use this record level keyword on the subfile control record format so that your program can clear the subfile of all records. This keyword differs from the SFLDLT keyword in that the subfile area is not deleted; it differs from the SFLINZ keyword in that after being cleared, the subfile contains no data. Clearing the subfile does not affect the display; however, after being cleared, the subfile contains no active records.

When active records already exist in the subfile and all are to be replaced, your program can issue an output operation to the subfile control record format after selecting the SFLCLR keyword. This clears the subfile and permits your program to write new records to the subfile (by issuing output operations to the subfile record format while incrementing the relative record number). Issuing an output operation to an already active subfile record causes an error message to be returned to your program.

If the SFLCLR keyword is in effect on an output operation and no records exist in the subfile, the SFLCLR keyword is ignored.

An option indicator is required for this keyword to prevent CPF from clearing the subfile on every output operation to the subfile control record format.

This optional keyword is valid only for the subfile control record format. Screen size condition names are not valid for this keyword.

The following example shows how to specify the SFLCLR keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A								R		SFLR								SFL
	A										(at least one displayable field)								
	A																		
00020	A								R		SFLCTLR								SFLCTL(SFLR)
00030	A																		SFLPAG(17)
00040	A																		SFLSIZ(17)
00050	A			01															SFLDSP
00060	A			01															SFLDSPCTL
00070	A	N		01															SFLCLR
	A																		
	A																		
	A																		

Normally, the option indicator(s) specified for the SFLCLR keyword is the reverse of the option indicator(s) specified for the SFLDSP and SFLDSPCTL keywords. In this example, the subfile is displayed when option indicator 01 is set on for an output operation to SFLCTLR, and the subfile is cleared when option indicator 01 is set off for an output operation to SFLCTLR.

SFLCTL(subfile-record-format-name)

Use this record level keyword (subfile control) to specify that this record format is to be a subfile control record format. This record format must immediately follow the subfile record format.

You must specify the name of the subfile record format as the parameter value for this keyword. The subfile control record format can contain field descriptions as well as subfile control keywords.

Your program can display subfile records only by issuing an output operation to the subfile control record format.

The subfile record format (SFL keyword) defines the format of the records in the subfile as opposed to the subfile control record format (SFLCTL keyword), which defines how the subfile can be displayed, cleared, deleted, and initialized. The program issues output operations to the subfile record format to build the subfile; it issues output operations to the subfile control record format, setting option indicators for various subfile keywords, to display, clear, delete, and initialize the subfile. (See SFLMSGRCD, SFLMSGKEY, and SFLPGMQ keyword descriptions for information on message subfiles.)

Option indicators are not valid with this keyword.

The following example shows how to specify the SFLCTL keyword:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blR/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blA/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blO/I/B/H/M)	Location: Line	Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		SFLR								SFL
	A																		
	A										(at least one displayable field)								
	A																		
00020	A								R		SFLCTLR								SFLCTL(SFLR)
00030	A																		SFLPAG(17)
00040	A																		SFLSIZ(17)
00050	A																		SFLDSP
	A																		
	A																		
	A																		

The parameter value on the SFLCTL keyword must be the name of the subfile record format (identified by the SFL keyword).

A summary of subfile keywords used with the SFLCTL keyword follows. (Field level keywords are used with fields in the subfile control record format):

Required	Optional
SFLCTL	SFLLIN
SFLPAG	SFLEND
SFLSIZ	SFLDSPCTL
SFLDSP	SFLCLR
	SFLDLT
	SFLDROP
	SFLENTER
	SFLINZ
	SFLRCDNBR
	SFLRNA
	SFLROLVAL
	SFLMSG
	SFLMSGID
	SFLPGMQ

If subfile size equals subfile page, the following keywords are ignored. When several display sizes are used (DSPSIZ keyword specified), these keywords are ignored only for screen sizes for which subfile size equals subfile page.

SFLDROP
SFLROLVAL

If the subfile record format contains field selection, the following keywords are not valid on the subfile control record format:

SFLDROP
SFLINZ
SFLLIN
SFLRCDNBR
SFLRNA (because SFLINZ is not valid)
SFLROLVAL

The USRDFN keyword is not valid for the subfile control record format.

SFLDLT

Use this record level keyword with an option indicator on the subfile control record format to enable your program to delete the subfile. When two subfiles in a display file are already active and a third subfile is to be made active, your program must delete one of the active subfiles before making the third active. To make a subfile active, your program issues an output operation to the subfile record format or issues an output operation to the subfile control record format with SFLINZ in effect. To delete a subfile, your program issues an output operation to the subfile control record format with SFLDLT in effect. (Closing the display file deletes all the active subfiles.)

If your program issues an output operation with SFLDLT in effect to a subfile that is not active, the SFLDLT keyword is ignored.

Option indicators are required for this keyword; screen size condition names are not valid.

The following example shows how to specify the SFLDLT keyword:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/N)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		SFLR								SFL
	A																		
	A										(at least one displayable field)								
	A																		
00040	A								R		SFLCTLR								SFLCTL(SFLR)
00050	A																		SFLPAG(17)
00060	A																		SFLSIZ(17)
00070	A			01															SFLDSP
00080	A			01															SFLDSPCTL
00090	A			04															SFLDLT
	A																		
	A																		
	A																		

Normally, the option indicator(s) specified for the SFLDLT keyword is different from the option indicator(s) specified for the SFLDSP and SFLDSPCTL keywords. In this example, the subfile is displayed when option indicator 01 is set on for an output operation to SFLCTLR, and the subfile is deleted when option indicator 04 is set on for an output operation to SFLCTLR.

SFLDROP(CAnn | CFnn)

Use this record level keyword on the subfile control record format to assign a CA or a CF key. The work station user presses this key to fold or to truncate subfile records that require more than one display line.

Without the SFLDROP keyword, CPF displays the entire subfile record and folds it where needed. When the SFLDROP keyword is specified, CPF first displays the subfile in truncated form; subfile records are truncated to fit on one display line. When the work station user presses the specified key, CPF redisplays the records in folded form. Each record continues onto subsequent lines immediately following the line the record starts on. See example 13 in Appendix F, *Examples*.

Pressing the specified key changes the form of the displayed subfile from one state to the other.

In the truncated form, more records are displayed than are specified on the SFLPAG keyword; in the folded form, as many records are displayed as are specified on the SFLPAG keyword.

CPF truncates subfile records in the middle of output-only fields; however, if the truncation is in the middle of an input-capable field, the whole field is omitted from the screen. If this results in omitting the entire record from the screen, an error message is sent to the display and the record is not truncated. Instead, it is displayed in folded form.

Notes:

1. A warning message is issued at file creation time if the entire record fits on a single screen line.
2. If subfile size equals subfile page, the SFLDROP keyword is ignored. When several display sizes are used (DSPSIZ keyword specified), SFLDROP is ignored only for screen sizes for which the subfile size equals subfile page. If the subfile record format contains field selection, the SFLDROP keyword is not valid.
3. If the subfile contains input-capable fields, it is recommended that you specify a CF key rather than a CA key. If you specify a CA key in this situation, modified data is lost when the key is pressed.
4. If two subfiles using SFLDROP are displayed at one time, the same command key should be specified on both SFLDROP keywords. If they are different, only the key specified for the most recently displayed subfile is in effect. Pressing the command key affects the subfile containing the cursor. If the cursor is not positioned in a subfile, the command key affects the upper subfile.

Option indicators are not valid with this keyword.

The following example shows how to specify the SFLDROP keyword:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		SFLR								SFL
	A																		
	A										(subfile records should not fit on one screen line)								
	A																		
00040	A								R		SFLCTLR								SFLCTL(SFLR)
00050	A																		SFLPAG(17)
00060	A																		SFLSIZ(34)
00070	A																		SFLDSP
00080	A																		SFLDROP(CF03)
	A																		
	A																		
	A																		

In this example, when the subfile is displayed, the work station user can press the CF03 key to change the subfile from truncated to folded form and from folded to truncated form.

SFLDSP

Use this record level keyword on the subfile control record format so that CPF displays the subfile when your program issues an output operation to the subfile control record format. If you do not use an option indicator with this keyword, a page of subfile records is displayed on every output operation to the subfile control record format.

See the SFLRCDNBR keyword description to determine which page of subfile records is displayed when the subfile is displayed.

If your program issues an output operation to the subfile control record format when the SFLDSP keyword is in effect and no records exist in the subfile, an error message is sent to your program.

This keyword is required, and is valid only for the subfile control record format. Screen size condition names are not valid for this keyword; option indicators are valid.

The following example shows how to specify the SFLDSP keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/N)	Decimal Positions	Usage (B/O/I/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		SFLR								SFL
	A																		
	A										(at least one displayable field)								
	A																		
00040	A								R		SFLCTLR								SFLCTL(SFLR)
00050	A																		SFLPAG(17)
00060	A																		SFLSIZ(17)
00070	A			01															SFLDSP
	A																		
	A																		
	A																		

In this example, the subfile is displayed when option indicator 01 is set on for an output operation to SFLCTLR.

SFLDSPCTL

Use this record level keyword on the subfile control record format so that CPF displays fields in the subfile control record format when your program issues an output operation to the subfile control record format. If you do not use an option indicator with this keyword, the subfile control record is displayed on every output operation to the subfile control record format.

This optional keyword is valid only for the subfile control record format. Screen size condition names are not valid for this keyword; option indicators are valid.

Note: SFLDSPCTL must be in effect when the subfile is displayed for an input operation to the subfile control record to be valid, even if there are no displayable fields in the subfile control record format.

The following example shows how to specify the SFLDSPCTL keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/N)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		SFLR								SFL
	A																		
	A										(at least one displayable field)								
	A																		
00040	A								R		SFLCTLR								SFLCTL(SFLR)
00050	A																		SFLPAG(17)
00060	A																		SFLSIZ(17)
00070	A																		SFLDSP
00080	A																		SFLDSPCTL
	A																		
	A																		
	A																		

In this example, both the subfile and displayable fields in the subfile control record format are displayed when option indicator 01 is set on for an output operation to SFLCTLR.

SFLEND

Use this record level keyword on the subfile control record format to permit the display of a plus sign (+) in the lower, rightmost display location occupied by the subfile. The plus sign indicates that the work station user can roll up the subfile to display more records by pressing the Roll Up key.

An option indicator *must* be specified with this keyword.

Rolling by Your Program (SFLPAG Equals SFLSIZ): Your program controls the display of the plus sign through the use of the indicators on the SFLEND keyword. Set the indicators off to display the plus sign; set the indicators on to remove the plus sign from the display. When the Roll Up key is pressed, your program handles processing; for instance, it reads the subfile, clears it, then rewrites the subfile with new records and redisplays the subfile. If your program does this, display the plus sign; if not, remove the plus sign from the display.

Rolling by CPF (SFLPAG Does Not Equal SFLSIZ): CPF displays the plus sign as long as there are more records in the subfile to be displayed, no matter how the option indicator is set. When the last page of the subfile is displayed, CPF displays the plus sign if the indicator is off and does not display the plus sign if the indicator is on.

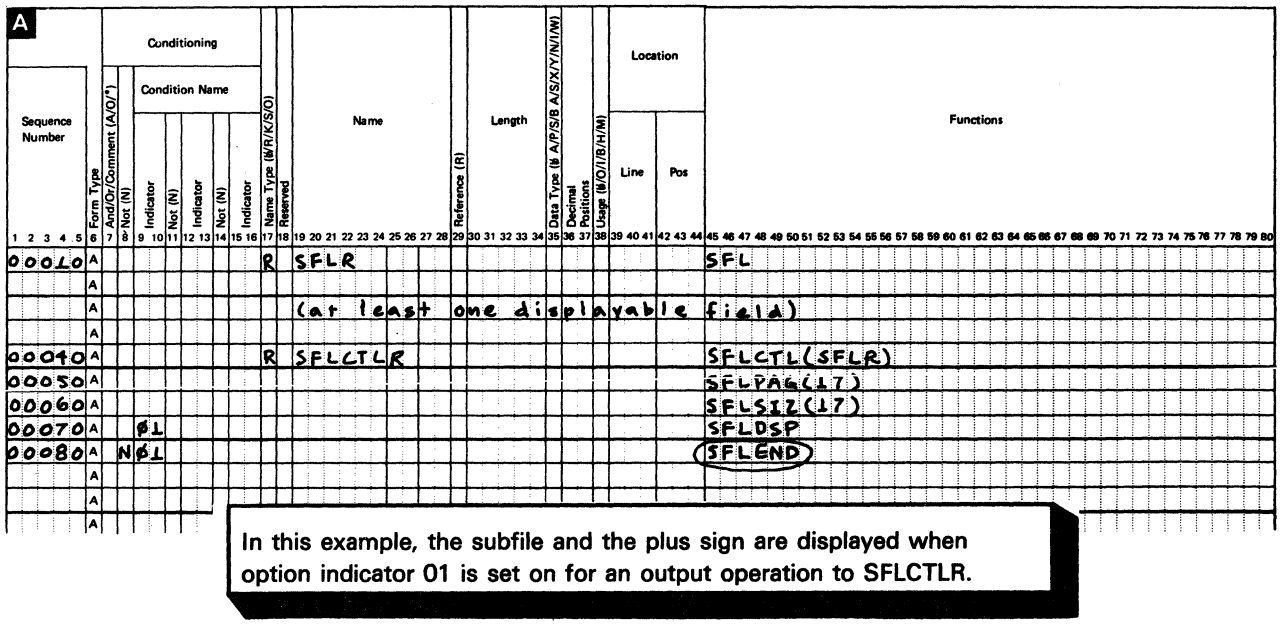
Your program must set the indicator on or off when *displaying the subfile.* (Your program cannot find out, when CPF is rolling the subfile, which page of the subfile is displayed.)

If your program sets off the indicator for SFLEND when displaying the subfile, the plus sign is displayed with the last page of the subfile. Because the plus sign is displayed but CPF cannot roll the subfile any further, your program must provide for any further rolling. Specify the ROLLUP keyword on the subfile control record format so that control is passed to your program when the Roll Up key is pressed again. When your program receives control, it must clear the subfile, refill it with new subfile records, and redisplay the first page of subfile records.

Position of Plus Sign: For the 5250 work station, positions 78 through 80 of the last line occupied by the subfile are used for the plus sign (beginning attribute character, plus sign, and ending attribute character). For the system console, positions 63 through 64 of the last line occupied by the subfile are used for the plus sign (screen attribute, plus sign).

Note: For both the 5250 work station and the system console, if an input field occupies the location of the plus sign and the field is modified, the plus sign and its attribute characters are returned to the program as data in the field.

The following example shows how to specify the SFLEND keyword:



Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A								R		SFLR								SFL
	A										(at least one displayable field)								
00040	A								R		SFLCTLR								SFLCTL(SFLR)
00050	A																		SFLPAG(17)
00060	A																		SFLSIZ(17)
00070	A			01															SFLDSP
00080	A		N	01															SFLEND
	A																		
	A																		
	A																		

In this example, the subfile and the plus sign are displayed when option indicator 01 is set on for an output operation to SFLCTLR.

SFLENTER(CAnn | CFnn)

Use this record level keyword on the subfile control record format to specify that the Enter/Rec Adv key is to be used as a Roll Up key. This allows CPF to roll up a subfile of more than one page when the Enter/Rec Adv key is pressed. The parameter value with this keyword is required; use it to specify a command key to replace the Enter/Rec Adv key while this function is active. This keyword is normally used when the subfile is entirely entered by the work station user or when the user changes some records in the subfile and adds others.

If a subfile is not currently displayed and the Enter/Rec Adv key is pressed, control is returned to your program.

This optional keyword is valid only for the subfile control record format. Option indicators are not valid for this keyword.

Note: This keyword is in effect only until the next output operation. At the next output operation, the specifications for that record apply.

If two subfiles with SFLENTER in effect are displayed, only the last SFLENTER keyword to be displayed is in effect. The cursor position at the time the key is pressed determines which subfile is affected.

The following example shows how to specify the SFLENTER keyword:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: And (N)	Conditioning: Condition Name: Or (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (M/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	9	10	11	12	13	14	15	16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36	37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A									R		SFLR								SFL
	A																			
	A											(at least one displayable field)								
	A																			
00040	A									R		SFLCTLR								SFLCTL(SFLR)
00050	A																			SFLPAG(17)
00060	A																			SFLSIZ(51)
00070	A																			SFLDSP
00080	A																			SFLENTER(CF01)
	A																			
	A																			
	A																			

In this example, the Enter/Rec Adv key is used as a Roll Up key; to enter data, the work station user presses CF01.

SFLINZ

Use this record level keyword on the subfile control record format to specify that CPF is to initialize all records in the subfile on an output operation to the subfile control record format (identified by the SFLCTL keyword). The fields in each subfile record are initialized to blanks for character type fields, to zeros for numeric type fields, or to the constant value specified on input-only fields for which the DFT keyword is specified.

When the subfile is displayed (on an output operation to the subfile control record), all records in the subfile are displayed with the same value. Any record previously written is overwritten and no longer has its earlier value.

The following is true when SFLINZ is in effect on an output operation to the subfile control record format. If keywords (such as DSPATR(HI)) are specified on fields in the subfile record format, and if option indicators are specified on those keywords, the subfile is displayed as if all option indicators are off (hex F0). Note that a keyword can be selected if N is specified for the option indicator.

After your program issues an output operation to the subfile control record with SFLINZ in effect, all records in the subfile are considered active but not modified. They are considered modified only when the work station user modifies them or when your program issues an output operation to the subfile record format with SFLNXTCHG in effect.

To initialize a subfile with no active records, see the SFLRNA keyword.

In general, use the SFLINZ keyword for the following purposes:

- Specify SFLINZ with UNLOCK(*ERASE *MDTOFF) when your program is to build a transaction file. Do not specify validity checking keywords for fields in the subfile record format. This is a fast data-entry approach that is discussed in the *Application Example I*.

- Specify SFLINZ with SFLRNA so that the work station user can add records to a subfile that your program has already partially filled with records before displaying it.

- Specify SFLINZ with SFLPGMQ so that your program can build a message subfile with a single output operation. (See the SFLPGMQ keyword description later in this chapter.)

Notes:

1. If field selection is used in the subfile record format, the SFLINZ keyword is not valid. Your program can only initialize the subfile by a series of output operations to the subfile record format, selecting fields as needed.
2. The SFLINZ keyword cannot be specified on the subfile control record format for a message subfile (see the SFLMSGRCD keyword) unless the SFLPGMQ keyword is also specified at the field level in the same subfile control record format.

Option indicators are valid with this keyword; screen size condition names are not valid.

For an example of SFLINZ and SFLPGMQ, see example 19 in Appendix F. *Examples*.

For an example of SFLINZ and SFLRNA, see example 12 in Appendix F. *Examples*.

The following example shows how to specify the SFLINZ keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: or Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B/A/S/X/Y/N/I/N)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12	13	14 15	16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		SFLR								SFL
	A																		
	A										(at least one displayable field)								
	A																		
00040	A								R		SFLCTLR								SFLCTL(SFLR)
00050	A																		SFLPAG(17)
00060	A																		SFLSIZ(17)
00070	A																		SFLDSP
00080	A																		SFLINZ
00090	A																		UNLOCK(*ERASE *MDTOFF)
	A																		
	A																		
	A																		

For more information on initializing subfiles, see the *CPF Programmer's Guide*.

SFLLIN(spaces)

Use this record level keyword on the subfile control record format to specify that the subfile is to be displayed as a horizontal subfile (having more than one column of records displayed). The parameter value specifies the number of spaces (including attribute characters) between columns of records.

For example, specifying the SFLLIN keyword causes a subfile of four records to be displayed as:

```
REC1   REC3   The parameter value for SFLLIN
REC2   REC4   specifies the number of spaces
              appearing between columns.
```

If SFLLIN is not specified, these records appear as:

```
REC1
REC2
REC3
REC4
```

If the subfile record format contains field selection, this keyword is invalid.

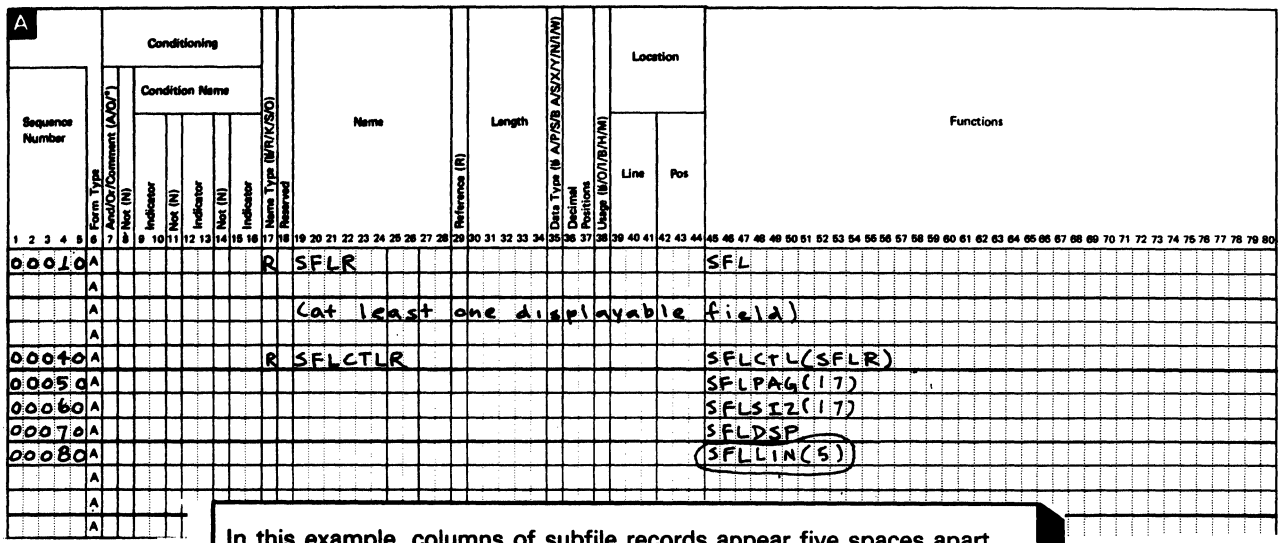
If the SFLLIN keyword is to be used for more than one screen size, a screen size condition name for each secondary screen size is required.

Because the SFLPAG keyword specifies the number of subfile records that can be displayed at a single time, you must consider the SFLLIN keyword when specifying the SFLPAG value.

The SFLLIN keyword is not valid for a message subfile.

Option indicators are not valid with this keyword.

The following example shows how to specify the SFLLIN keyword:



SFLMSG('message-text' [response-indicator])
SFLMSGID(message-identifier message-file[.library-name])

Use these record level keywords on the subfile control record format to identify a message to be displayed on the message line when your program does an output operation to the subfile control record format. Your program has the responsibility to reverse the images of any fields and to position the cursor appropriately in the subfile being displayed. As with ERRMSG and ERRMSGID, the cursor blinks and the keyboard is locked on the 5250 work station until the Reset key is pressed. (These conditions cannot be selected for the system console.)

SFLMSG keyword: Specify the SFLMSG keyword like the ERRMSG keyword. The parameters specify a message text and, optionally, a response indicator. The message text is the message to be displayed. (The Help key is not supported, and if pressed, it causes an error message to be sent to the display.)

If you specify a response indicator, it should be the same as the option indicator used to condition the SFLMSG keyword. On the input operation that follows the display of the error message, CPF turns off the indicator. If the response and option indicators are the same, they are both turned off. One exception to this rule is if the response indicator is also specified for another keyword such as CHANGE, CAnn, or CFnn. In that case, the on/off setting of the response indicator is based on the results of the function provided by the CHANGE or CFnn keyword.

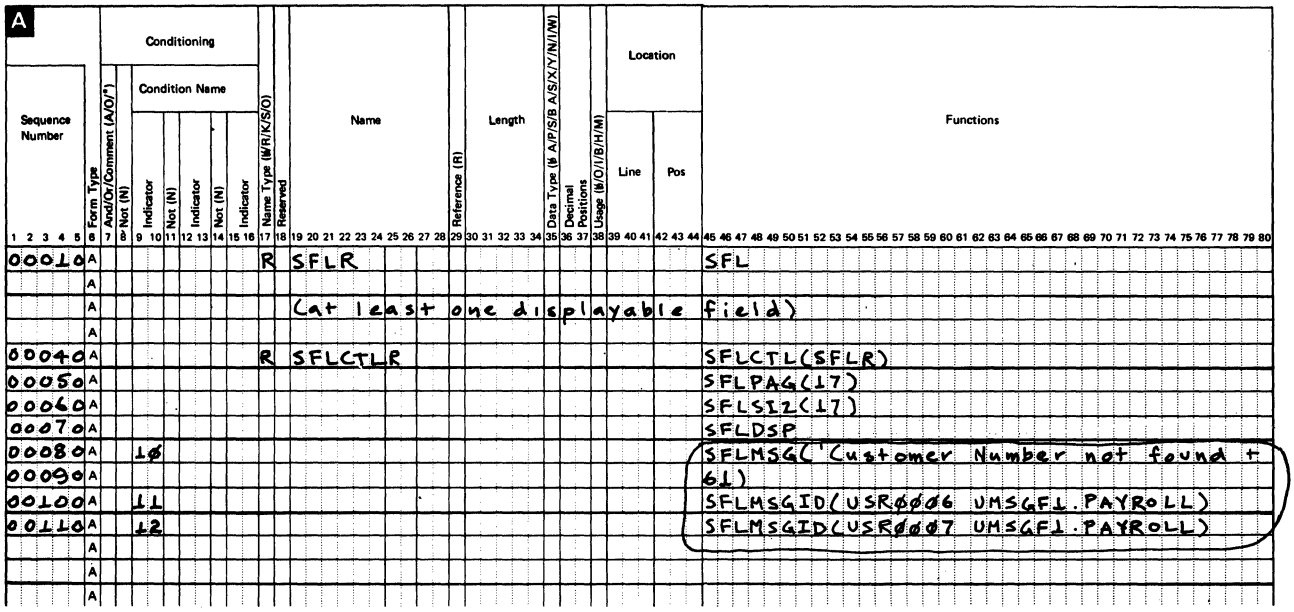
When a response indicator is specified, the first 50 characters of the message text are also used as indicator text. Separate response indicator text is not valid for the SFLMSG keyword.

SFLMSGID keyword: Specify the SFLMSGID keyword as you would the ERRMSGID keyword. The parameters specify a message identifier and a message file. The message to be displayed is retrieved from the message file during program execution. (The Help key is supported for this keyword and causes a predefined second-level message from the message file to be displayed if it exists.) The library name on the message file is optional.

Note: The message file used in SFLMSGID must exist in the specified library, and the user must have operational rights to the message at program execution time.

Option indicators are valid with these keywords.

The following example shows how to specify the SFLMSG and SFLMSGID keywords:



Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		SFLR								SFL
	A																		
	A										(at least one displayable field)								
	A																		
00040	A								R		SFLCTLR								SFLCTL(SFLR)
00050	A																		SFLPAG(17)
00060	A																		SFLSIZ(17)
00070	A																		SFLDSP
00080	A			10															SFLMSG('Customer Number not found + 61)
00090	A																		
00100	A			11															SFLMSGID(USR0006 UMSGF1.PAYROLL)
00110	A			12															SFLMSGID(USR0007 UMSGF1.PAYROLL)
	A																		
	A																		
	A																		

Whether to Specify Error Message Keywords (ERRMSG and ERRMSGID) or Subfile Message Keywords (SFLMSG and SFLMSGID):

- For fields in the subfile control record format
- For fields in any nonsubfile record format

CPF reverses the image of the field or fields in error and positions the cursor at the first such field.

Specify SFLMSG and SFLMSGID:

- At the record level in the subfile control record format for fields in the subfile record format

Your program must reverse the image of the field or fields in error (by selecting DSPATR(RI)) and position the cursor at the field to be corrected first (by selecting DSPATR(PC)).

Conditions Occurring During Message Display: The display of messages using SFLMSG and SFLMSGID is similar to the display of messages by CPF when field validation errors are detected. An important difference from ERRMSG and ERRMSGID is that the program, and not CPF, must position the cursor to the appropriate field within the subfile, reverse the image of that field within the subfile, and also optionally reverse the image of more than one field at a time. On the 5250 work station, blinking cursor and message highlighting are allowed. On the system console, these display attributes are not allowed.

Restoration of Reversed Image Fields: See *Restoration of Reversed Image Fields* in the ERRMSG/ERRMSGID keyword description in this chapter.

Note: The SFLDSP keyword must be in effect for SFLMSG and SFLMSGID to be processed.

Priority Among Selected Keywords: You can specify either SFLMSG or SFLMSGID several times for a single subfile control record format. During program execution, set option indicators to select a particular message to be displayed and to select particular fields to be displayed in reverse image. Several fields can be displayed in reverse image in different records of a subfile when the subfile is redisplayed; however, only one message can be displayed at one time.

If more than one error message is selected at a time, CPF displays the first of the following:

1.	ERRMSG (If more than one ERRMSG keyword is selected, the first one selected is displayed.)	Effective for individual fields within the subfile control record format
2.	ERRMSGID (If more than one ERRMSGID keyword is selected, the first one selected is displayed.)	
3.	SFLMSG (If more than one SFLMSG keyword is selected, the first one selected is displayed.)	Effective for program-selected fields in the displayed subfile (subfile record format)
4.	SFLMSGID (If more than one SFLMSGID keyword is selected, the first one selected is displayed.)	
5.	Message fields (M in position 38) (If more than one message field is selected, the first one selected is displayed.)	Effective when specified in the subfile control record format

SFLMSGKEY

Use this field level keyword on the first field in the subfile record format for a message subfile (SFL and SFLMSGRCD specified). It is not valid on the subfile control record format (SFLCTL). To select messages from a program message queue for display, your program places a message reference key in this field. (Your program also places the name of the program message queue in the second field in the subfile record format; see the SFLPGMQ keyword description).

This field is predefined as a four-position, character data type, hidden field. The following rules apply:

- This field must always be the first field defined in the subfile record format.
- The field name and the SFLMSGKEY keyword are the only DDS you can specify for this field.

Option indicators are not valid with this keyword or with the associated field.

For more information on building and displaying message subfiles, see the SFLPGMQ keyword description in this chapter.

The following example shows how to specify the SFLMSGKEY keyword:

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (7)	Not (8)	Indicator (9-10)	Not (11)	Indicator (12-13)	Not (14)	Indicator (15-16)	Name Type (17)	Reserved (18)	Name (19-28)	Reference (29)	Length (30-34)	Data Type (35)	Decimal Positions (36-37)	Usage (38)	Line (39-41)	Pos (42-44)	Functions (45-80)
00010	A								R		RCDMSG								SFL SFLMSGRCD(3)
00020	A										FLDKEY								SFLMSGKEY
00030	A										FLDPGM								SFLPGMQ
00040	A								R		SFLCTL								SFLCTL(RCDMSG)
00050	A			Ø1															SFLINZ
	A										⋮								⋮
00090	A										FLDPGM								SFLPGMQ
	A																		
	A																		
	A																		

SFLMSGRCD(line-number)

Use this record level keyword on the subfile record format to specify that this subfile is to be a *message subfile* and that the records displayed when the subfile is displayed are to be messages from a program message queue.

The parameter value specified with the SFLMSGRCD keyword specifies the first line on the display on which messages are displayed. The value specified must not be greater than the maximum line number for the screen size being used. The number of messages displayed depends on the SFLPAG value specified for the subfile.

For more information on building and displaying message subfiles, see the SFLPGMQ keyword description in this chapter.

Option indicators are not valid with this keyword; however, screen size condition names are valid with this keyword.

The TEXT keyword is valid at the record level for SFLMSGRCD and at the field level for the SFLMSGKEY and SFLPGMQ fields.

The following example shows how to specify the SFLMSGRCD keyword:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	A																		
00030	A								R		RCDMSG								SFL SFLMSGRCD(3)
00040	A										FLDKEY								SFLMSGKEY
00050	A										FLDPGM								SFLPGMQ
00060	A								R		SFLCTL								SFLCTL(RCDMSG)
	A																		
	A																		
	A																		

In this example, the shaded keywords are required on the subfile record format for a message subfile. SFLMSGKEY and SFLPGMQ must be specified in the order shown.

There can be only two predefined fields specified on the subfile record format for a message subfile as follows:

- *Message identifier*: A four-position, character data type, hidden field. Your program uses this field to pass a message identifier to CPF. This field must always be the first field defined in the message subfile. You must specify the SFLMSGKEY keyword with this field.

- *Program queue name*: A 10-position, character data type, hidden field. Your program passes the name of the program message queue that contains the message(s) in this field. It must be the second field of a subfile message record and must immediately follow the first field. If specified also on the subfile control record, it can be anywhere within the record specification. You must specify SFLPGMQ with this field. (See the SFLPGMQ keyword for more details.)

Screen size condition names can be specified for the SFLMSGRCD keyword and are required if the line number for the first message displayed is to change, based on display size.

Data is not returned in your input buffer if your program does an input operation for a message subfile.

The messages are displayed as follows:

- Each message is displayed on a separate line and is truncated if it is longer than the display line length.

- Each message starts in position 2. The maximum message length for the 5250 work station is 76 characters; the maximum for the system console is 61 characters.

- On the 5250 work station, each message is displayed with the high intensity (HI) field attribute.

When a message subfile is rolled by CPF, the cursor is positioned at the same location as it was when the Roll key was pressed.

Second-level text is supported for these messages. The work station user chooses which second-level text is to be displayed by placing the cursor on the line containing the first-level message and pressing the Help key.

Note: When the SFLMSGRCD keyword is specified, the SFLINZ keyword cannot be specified without the SFLPGMQ specification.

SFLNXTCHG

Use this record level keyword on the subfile record format to force the work station user to correct program-detected keying errors in subfile records that have been read by the program. It does this by causing a record to be modified so that a Get-Next-Changed operation must read the record as described in the following section.

Subfile Operations With SFLNXTCHG: A typical use of SFLNXTCHG could be as follows. A work station user changes (modifies) some records in a displayed subfile (this could be for a data-entry application or a data-update application). After changing some records, the work station user presses the Enter/Rec Adv key, and the program reads only the changed records (with Get-Next-Changed operations). If the program detects keying errors in the changed records, it can issue update operations (UPDAT in RPG; REWRITE in COBOL) to the subfile records in error, setting indicators if necessary so that SFLNXTCHG will be in effect during the update operations. These update operations are issued to the subfile record format. After the records in error have been updated, the program issues an output operation to the subfile control record format to redisplay the subfile. (To display an error message, display fields in reverse image, and position the cursor, see the keyword description for the SFLMSG and SFLMSGID keywords.) With the subfile redisplayed, the work station user rekeys data and presses the Enter/Rec Adv key again. If the data is correct, the program does not redisplay the subfile.

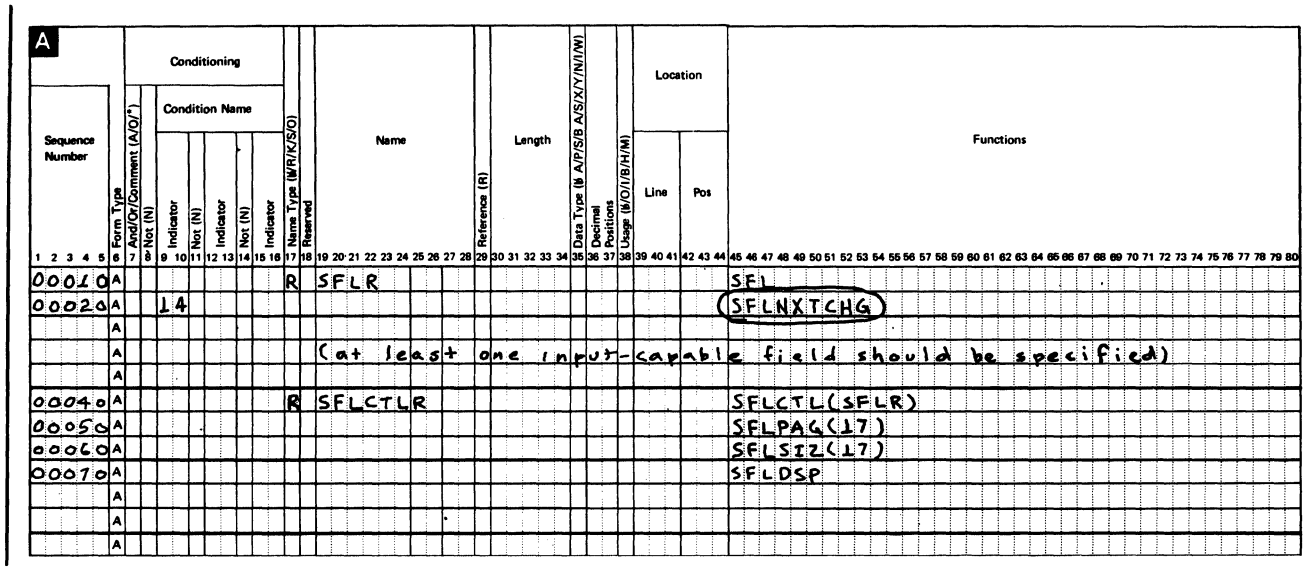
The records in error are returned to the program on a Get-Next-Changed operation. This is because the SFLNXTCHG keyword caused the subfile records to be considered modified even though the work station user did not change them. This allows the program to prohibit the work station user from ignoring program-detected keying errors in subfile records.

Subfile Operations Without SFLNXTCHG: If SFLNXTCHG is not specified, or is specified but not selected on the update operations to the subfile records, then the work station user can simply press the Enter/Rec Adv key instead of correcting the program-detected errors. The program then reads no records because the Get-Next-Changed operations find no modified records the second time the Enter/Rec Adv key is pressed.

Option indicators are valid with this keyword.

The SFLNXTCHG keyword cannot be specified with the SFLMSGRCD keyword.

The following example shows how to specify the SFLNXTCHG keyword:



A	Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (M/O/I/B/H/M)	Location: Line	Pos	Functions
	00010	A								R		SFLR								SFL
	00020	A			14															SFLNXTCHG
		A																		
		A										(at least one input-capable field should be specified)								
		A																		
	00040	A								R		SFLCTLR								SFLCTL(SFLR)
	00050	A																		SFLPAG(17)
	00060	A																		SFLSIZ(17)
	00070	A																		SFLDSP
		A																		
		A																		
		A																		

SFLPAG(number-of-records-to-be-displayed)

Use this record level keyword on the subfile control record format to specify the number of records in the subfile to be displayed at the same time. (For an exception to this rule, see *Field Selection* later in this keyword description.)

This keyword is required for the subfile control record format.

The SFLPAG parameter value and the number of lines required by each subfile record determine the number of actual display lines required to display the page of records. Not all records within a subfile must be displayed at the same time, and not all lines of the display are required to display a page of subfile records.

Option indicators are not valid for this keyword; screen size condition names are valid. Screen size condition names are required if you want the number of records that can be displayed at one time to change, based on the size of the display screen.

Subfile Page Equals Subfile Size

When you specify the same parameter values for the SFLPAG keyword and the SFLSIZ keyword, the maximum number of records that can be contained in the subfile equals the maximum number of subfile records that can appear on the display at one time. For this condition, CPF does not automatically roll the subfile when the Roll Up or the Roll Down key is pressed. If the ROLLUP and ROLLDOWN keywords are specified and one of the Roll keys is pressed, CPF returns control to your program instead. If ROLLUP and ROLLDOWN are not specified, a message is sent to the work station user, indicating that an invalid key has been pressed.

If subfile size equals subfile page, the following keywords are ignored. When several display sizes are used (DSPSIZ keyword specified), these keywords are ignored only for screen sizes for which subfile size equals subfile page.

SFLDROP
SFLROLVAL

Field Selection: When subfile page equals subfile size, you can specify option indicators for fields in the subfile record format. This is called *field selection.* When field selection is used in the subfile record, SFLPAG(value) specifies the number of display lines available to display the records of this subfile. (Without field selection, SFLPAG(value) specifies the number of subfile records that can be displayed at one time.) This specification must be considered when a subfile record occupies more than one display line. The value of SFLPAG must be greater than or equal to the number of display lines occupied by the subfile.

If the subfile record format contains field selection, the following keywords are not valid on the subfile control record format:

SFLDROP
SFLINZ
SFLLIN
SFLRCDNBR
SFLRNA (because SFLINZ is not valid)
SFLROLVAL

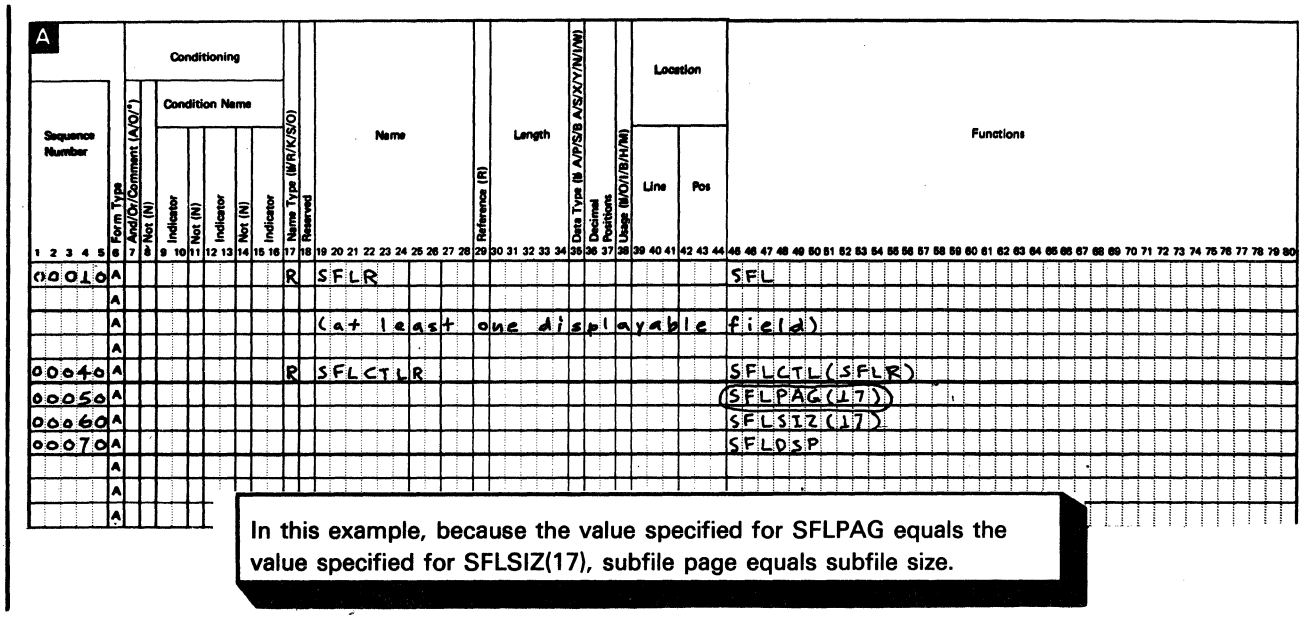
Option indicators for the following field level keywords associated with the subfile record format are valid only if the subfile size equals a subfile page:

AUTO(RA)
CHECK(ER)
CHECK(ME)
DSPATR(MDT)
DSPATR(PR)
DUP

Subfile Page Does Not Equal Subfile Size

When you specify different parameter values for the SFLPAG keyword and the SFLSIZ keyword, CPF recognizes the Roll Up and Roll Down keys and automatically rolls the subfile according to the value specified in the field for which the SFLROLVAL keyword is specified. If you do not specify the SFLROLVAL keyword, CPF rolls the subfile by the parameter value specified for the SFLPAG keyword.

The following example shows how to specify the SFLPAG keyword:



Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/Z/Y/N/I/M)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Pos	Functions
00010	A								R		SFLR								SFL
	A																		
	A										(at least one displayable field)								
	A																		
00040	A								R		SFLCTLR								SFLCTL(SFLR)
00050	A																		SFLPAG(17)
00060	A																		SFLSIZ(17)
00070	A																		SFLDSP
	A																		
	A																		
	A																		

In this example, because the value specified for SFLPAG equals the value specified for SFLSIZ(17), subfile page equals subfile size.

SFLPGMQ

Use this field level keyword on the second (and last) field in the subfile record format for a message subfile. This field contains the name of the program message queue used by CPF to build a message subfile. In addition, the SFLPGMQ keyword can be specified on the subfile *control* record format when the SFLINZ keyword is specified on the subfile control record format.

This field is predefined as a 10-position, character data type, hidden field. The following rules apply:

- The field name and the SFLPGMQ keyword are the only DDS you can specify for this field.
- If the name of the program message queue name placed in this field at execution time is less than 10 positions, it must be left-adjusted and padded with blanks.

This field is required on the subfile record format (identified by the SFL keyword) to build the subfile one message at a time through multiple output operations to the subfile record format. You can also specify this field on the subfile control record format (identified by the SFLCTL keyword) to build the subfile all at once through a single output operation to the subfile control record. Specify option indicators with the SFLINZ keyword to control the way the subfile is built.

Multiple Output Operation: If you specify this field and keyword on the subfile record, you build the subfile one message at a time with separate output operations to the subfile record format. For each output operation, the message identifier must be in the first field of the record (SFLMSGKEY keyword), and the name of the program message queue must be in the second field. At the time of the output operation, CPF retrieves the identified message from the queue and places it in the subfile as a record.

Single Output Operation: If you specify the SFLPGMQ keyword (with its named field) and the SFLINZ keyword on the subfile control record format, you build the entire subfile with one output operation directed to the subfile control record format. On the output operation, CPF initializes the subfile with all messages that are on the program message queue whose name is in the SFLPGMQ field. If necessary, CPF extends the subfile to contain all messages on the queue. For this function, the SFLMSGRCD, SFLMSGKEY, and SFLPGMQ keywords must be specified with the subfile record format (SFL keyword). The SFLPGMQ and SFLMSGKEY keywords are ignored for this function and your program need not set the values of their fields.

Special Value: The SFLPGMQ field can contain a special value, * (asterisk), instead of a program message queue name. If the program moves an asterisk to the SFLPGMQ field, CPF uses the message queue of the program issuing the output operation.

Both Multiple and Single Output Operations: If you specify the SFLPGMQ keyword with both the subfile record format and subfile control record format, you can use the single operation function one time and the multiple operation function some other time. Do this by setting indicators before issuing the output operation; however, all operations to a particular subfile must be consistent (multiple or single, but not intermixed) when preparing for a single display of the subfile.

Option indicators and screen size condition names are not valid for this keyword.

Note: An update operation to the message subfile is valid regardless of how the subfile is built.

The following example shows how to specify the SFLPGMQ keyword:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		RCDMSG								SFL SFLMSGRCD(3)
00020	A										FLDKEY								SFLMSGKEY
00030	A										FLDPGM								(SFLPGMQ)
00040	A								R		SFLCTL								SFLCTL(RCDMSG)
00050	A			01															SFLINZ
	A										↑								↑
00090	A										FLDPGM								(SFLPGMQ)
	A																		
	A																		
	A																		

In this example, the program can build the subfile with more than one output operation (indicator 01 off) or a single output operation (indicator 01 on) to the subfile control record format.

SFLRCDNBR[(CURSOR)]

Use this field level keyword on the subfile control record format to specify that the first page of the subfile to be displayed is to be the page containing the record whose relative record number is in this field. If you do not specify this keyword, CPF displays the first page of the subfile by default.

If CURSOR is specified, the cursor is placed in the subfile record whose relative record number is identified by the contents of this field. The cursor is positioned at the first input-capable field in the subfile record; if there is no input-capable field, the cursor is positioned at the first output-only or constant field. For example, if a page can contain three records, and nine records are contained in the subfile, a SFLRCDNBR field value of 8 causes records 7, 8, and 9 to be displayed. If CURSOR is specified, the cursor appears in record 8.

This field must be a zoned decimal field with zero decimal positions. It must have the keyboard shift attribute of signed numeric (S in position 35), and it can be up to 4 digits in length. It must be defined as an output-only, an output/input, or a hidden field. If a value less than 1 or a value greater than the number of records in the subfile is contained in this field on an output operation to the subfile control record format, an error is returned to your program.

This optional keyword is valid only for the subfile control record format.

Option indicators are not valid for this keyword.

You cannot specify both SFLRCDNBR and SFLROLVAL for the same field.

If the subfile record format contains field selection, this keyword is invalid.

The following example shows how to specify the SFLRCDNBR keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (W A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (W/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		SFLR								SFL
	A																		
	A										(at least one displayable field)								
	A																		
00040	A								R		SFLCTLR								SFLCTL(SFLR)
00050	A																		SFLPAG(17)
00060	A																		SFLSIZ(17)
00070	A																		SFLDSP
00080	A										DSPREC		4		0B		4	12	SFLRCDNBR(CURSOR)
	A																		
	A																		
	A																		

In this example, either the program or the work station user can set the value of the field before displaying the subfile.

SFLRNA

Use this record level keyword (subfile records not active) with the SFLINZ keyword on the subfile control record format so that your program can initialize a subfile with no active records. To do this, your program issues an output operation to the subfile control record format with the SFLINZ keyword selected. The subfile itself becomes active; the subfile records are not considered active unless one of the following occurs:

- Your program issues an output operation to the subfile record format, placing data in one of the subfile records. The subfile record becomes active but is not considered modified unless the SFLNXTCHG keyword is also in effect.

- After your program displays the subfile, the work station user keys data into subfile records. The records keyed into become active and modified.

The SFLRNA keyword is normally used so that a program can write some records to the subfile before displaying it and the work station user can then add records to the subfile.

When your program displays a subfile initialized with the SFLINZ and SFLRNA keywords in effect, fields in inactive records have the following values:

- Character fields are blank.

- Numeric fields are zero-filled.

- An input-only field with a constant value specified has the constant value.

Your program cannot issue an input operation to an inactive subfile record. Issuing a Get-Next-Changed operation to one of the subfile records returns the record only when the record has become active and been modified.

Your program cannot issue output operations to active records (SFLRNA not specified); it must issue update operations. Also, your program cannot issue update operations to inactive records (SFLRNA specified); it must issue output operations.

Notes:
1. The SFLINZ keyword is required when the SFLRNA keyword is specified.
2. The SFLRNA keyword cannot be specified for a message subfile (identified by the SFLMSGRCD keyword on the subfile record format).
3. If the subfile record format contains field selection, the SFLRNA keyword is not valid.

Option indicators are not valid with this keyword.

The following example shows how to specify the SFLRNA keyword:

Sequence Number	Form Type	Name Type	Name	Functions
00010	A	R	SFLR	SFL
	A			
	A		(at least one displayable field)	
	A			
00040	A	R	SFLCTLR	SFLCTL(SFLR)
00050	A			SFLPAG(17)
00060	A			SFLSIZ(17)
00070	A			SFLDSP
00080	A			SFLINZ
00090	A			SFLRNA
	A			
	A			
	A			

SFLROLVAL

Use this field level keyword in the subfile control record format to specify that the work station user can key a value into this field to tell CPF how many records to roll up or down when the appropriate Roll key is pressed.

This field must have the keyboard shift attribute of signed numeric with zero decimal positions. It can be up to 4 digits in length, and it must be defined as an output/input or input-only field.

You cannot specify both the SFLROLVAL and the SFLRCDNBR keywords for the same field.

The work station user can roll the data being displayed up or down by first keying in the number of records to roll by, then pressing the Roll Up or the Roll Down key. (On subsequent rolls, the SFLROLVAL value stays the same unless a new number is keyed in before rolling.) If a negative number or zero is keyed into this field and a Roll key is pressed, an error message is displayed at the work station.

This keyword is valid only for the subfile control record format; you must specify it if CPF is to support the roll-by-record function.

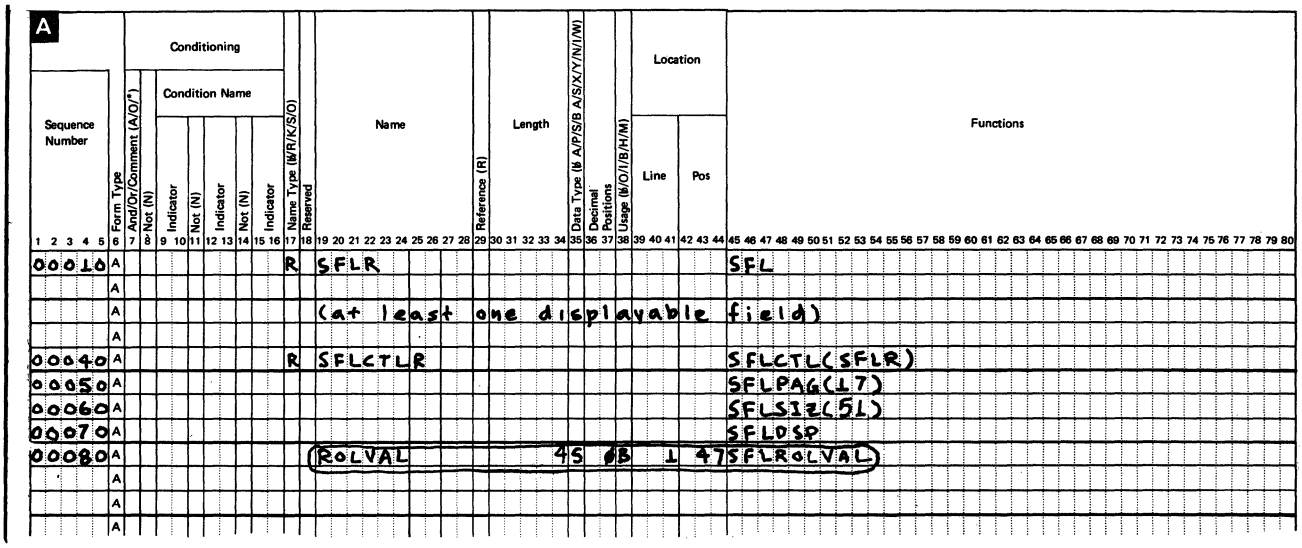
If this keyword is not specified, CPF rolls the display by the SFLPAG value.

If subfile size equals subfile page, the SFLROLVAL keyword is ignored. When several display sizes are used (DSPSIZ keyword specified), SFLROLVAL is ignored only for screen sizes for which subfile size equals subfile page. If the subfile record format contains field selection, the SFLROLVAL keyword is not valid.

This field is returned to your program as part of the input for this subfile control record.

Option indicators are not valid with this keyword.

The following example shows how to specify the SFLROLVAL keyword:



Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		SFLR								SFL
	A																		
	A										(at least one displayable field)								
	A																		
00040	A								R		SFLCTLR								SFLCTL(SFLR)
00050	A																		SFLPAG(17)
00060	A																		SFLSIZ(51)
00070	A																		SFLDSP
00080	A										ROLVAL		4	S	0	B	1	47	SFLROLVAL
	A																		
	A																		
	A																		

If pressing the Roll Down key rolls beyond the first page of records of the subfile, two conditions can occur. First, if the first page of records is not currently displayed, rolling down will display it. Second, if the first page of records is currently displayed, rolling down will cause a message to be displayed. (See below for what happens if ROLLDOWN is specified.)

If pressing the Roll Up key rolls beyond the last active record of the subfile, two conditions can occur. First, if the last full page of records is not already displayed, keying Roll Up will display it. Second, if the last full page of records is already displayed, keying Roll Up will cause a message to be displayed. *One exception to this rule is that when the SFLROLVAL value is less than the SFLPAG value, CPF rolls the subfile again and no message is displayed.*

Certain keywords are helpful when specified with SFLROLVAL:

- The SFLEND keyword notifies the work station user when the last subfile record is displayed. For more information on the SFLEND keyword, see that keyword description in this chapter.

- The ROLLUP or ROLLDOWN keywords cause control to return to the program when pressing a Roll Up or a Roll Down key would roll beyond the end of the subfile. Without ROLLUP or ROLLDOWN, a message is displayed (as described previously).

The following examples illustrate the use of the SFLROLVAL keyword:

- *Rolling Down*:

  Assume that the value specified for SFLPAG is 3, and there are 11 active records in the subfile. If records 8 through 10 are currently being displayed, and the user enters a roll value greater than 7, pressing the Roll Down key displays records 1 through 3.

  If records 1 through 3 are currently displayed, and a Roll Down key is entered with a SFLROLVAL value greater than 0, either a message is sent to the work station user (ROLLDOWN not specified) or control is returned to the user program (ROLLDOWN specified; the program has responsiblity for rolling down).

- *Rolling Up*:

  Assume that the value specified for SFLPAG is 3, and there are 11 active records in the subfile. If records 8 through 10 are currently being displayed, and the user enters a 3 into the SFLROLVAL field, pressing the Roll Up key displays record 11 in the uppermost page area of the display. Any lines not occupied by that record are blank. If the Roll Up key is pressed again, the last full page of subfile records (records 9 through 11) are displayed. Finally, if the Roll Up key is pressed a third time, either a message is sent to the work station user (ROLLUP not specified) or control is returned to the user program (ROLLUP specified; the program has responsibility for rolling up).

This chart lists the conditions that occur when rolling *beyond the ends of the subfile* (when the SFLROLVAL value is greater than SFLPAG value):

On pressing the Roll Up key: If the last full page of records is not already displayed, then it is displayed. If the last full page of records is already displayed, then a message is displayed. **On pressing the Roll Up key a second time:** If ROLLUP is specified, control returns to your program. If ROLLUP is not specified, a message is displayed.
On pressing the Roll Down key: If the first page of the subfile is not already displayed, then it is displayed. If the first page of the subfile is already displayed, then: If ROLLDOWN is specified, control returns to your program. If ROLLDOWN is not specified, a message is displayed.

Note: See the *CPF Programmer's Guide* for an explanation of how the position of the cursor at the time the Roll key is pressed affects rolling.

SFLSIZ(number-of-records-in-subfile)

Use this record level keyword on the subfile control record format to specify the number of records in the subfile. The maximum number of records allowed is 9999.

This keyword is required for the subfile control record format.

Option indicators are not valid with this keyword. Screen size condition names are valid and are required if the number of records within the subfile changes depending on the screen size.

Subfile Size Equals Subfile Page: When you specify the same parameter values for the SFLSIZ keyword and the SFLPAG keyword, you can specify option indicators for fields in the subfile record format. (This is called field selection.) When the subfile is built, the records can vary in length depending on which fields are selected, and each output operation places records into successive positions within the subfile. When the subfile is displayed, each record can require a different number of display lines. The number of records that actually fit in the subfile depends on the fields selected for each record written to the subfile. If the last subfile record written to the subfile fits exactly into the subfile, a status message (CPF 5003) is returned to your program. If the last subfile record written to the subfile overflows the subfile, a notify message (CPF 5043) is returned to your program.

The specified SFLPAG value is incremented to equal the maximum number of records that fit on the display when all the following conditions are true:

- The value of SFLSIZ equals the value of SFLPAG.
- Field selection is not used.
- SFLPAG and SFLLIN are specified such that the number of subfile records to be displayed cannot occupy a full screen.

To maintain SFLSIZ(value) equal to SFLPAG(value), the SFLSIZ value is incremented to equal the SLFPAG value.

For example, if SFLPAG(13) and SFLSIZ(13) are specified, and the subfile record format and SFLLIN value are specified such that three records can fit on a single display line, SFLPAG and SFLSIZ are incremented to 15.

Consider the following example. You specify the DDS for a subfile like this:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blR/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (b A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (bl/O/I/B/H/M)	Location: Line	Pos	Functions
1 2 3 4 5	6	7		9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		SFLR								SFL
00020	A			21							FIELD1		78				2	2	
00030	A			22							FIELD2		40				3	2	
00040	A	*																	
00050	A								R		SFLCTLR								SFLCTL(SFLR)
00060	A																		SFLPAG(5)
00070	A																		SFLSIZ(5)
00080	A																		SFLDSP
00090	A																		SFLDSPCTL
	A																		
	A																		

Your program issues the following output operations:

Output Operation to	Option Indicators Set	Result
SFLR	21 on 22 off	Only FIELD1 written to subfile
SFLR	21 on 22 on	FIELD1 and FIELD2 written to subfile
SFLR	21 off 22 on	Only FIELD2 written to subfile

(CPF issues status message CPF 5003 to your program.)

SFLCTLR	no indicator necessary	Subfile displayed

The resulting display looks like this:

A fourth record cannot be written to the subfile because there is not room on the display for it (SFLPAG(5) has been specified in the DDS).

Subfile Size Does Not Equal Subfile Page: When you specify different parameter values for the SFLPAG keyword and the SFLSIZ keyword, the SFLSIZ value specifies the number of records that can be placed into the subfile. If your program places a record with a relative record number larger than the SFLSIZ value into the subfile, the subfile is automatically extended to contain it (up to a maximum of 9999 records).

SLNO(n | *VAR)

Use this record level keyword (starting line number) to specify a starting line number for the record format you are defining. The SLNO keyword, if specified, adjusts the actual line numbers for each field in the record format. If you do not specify SLNO, fields in the record format are displayed on the lines you specify for them in positions 39 through 41. You can specify one of two parameter values for this keyword:

- Specify n, where n is a value between 1 and 24. All fields in the record format are offset n - 1 lines down the screen from their specified locations.

- Specify *VAR to enable your program to set the starting line number at execution time before displaying the record format. At file creation time, CPF sets the starting line number to one. If your program does not set the starting line number or sets it to zero, CPF assumes its value is one. If your program sets the starting line number to a value such that the first field in the record format does not all fit on the screen, or sets it to a negative value, CPF issues a notify message (CPF 5002) to your program, and the record is not displayed.

  To calculate the line on which a field is actually displayed, subtract one from the line number specified in positions 39 through 41 and add the starting line number to the result.

If the CLRL(nn) keyword is also in effect for this record when it is to be displayed, lines on the screen are cleared beginning with the starting line number.

The following keywords are ignored if the starting line number has changed since the last output operation:

ERRMSG
ERRMSGID
PUTOVR
PUTRETAIN

The SLNO keyword is invalid in a record format that has only hidden or message fields or that has no fields specified. It is also invalid in a record format that has one of the following keywords specified:

ASSUME
KEEP
SFL
SFLCTL
USRDFN

The SLNO keyword cannot be specified for the record format specified by the PASSRCD keyword.

Option indicators are not valid for this keyword.

The following example shows how to specify the SLNO keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		RECORD1								SLNO(*VAR)
00020	A										FIELD1		5			I	2	2	
00030	A										FIELD2		5			B	3	2	
00040	A																		
00050	A								R		RECORD2								SLNO(2)
00060	A										FIELD3		5				10	2	
00070	A										FIELD4		5			B	10	15	
	A																		
	A																		
	A																		

In this example, when the starting line number is zero or one, FIELD1 is displayed on line 2 and FIELD2 is displayed on line 3, as specified. When the starting line number is set to 2 in your program, FIELD1 is displayed on line 3 (2 - 1 + 2 = 3) and FIELD2 is displayed on line 4 (2 - 1 + 3 = 4).

FIELD3 and FIELD4 are always displayed on line 11 (2 - 1 + 10 = 11).

TEXT('description')

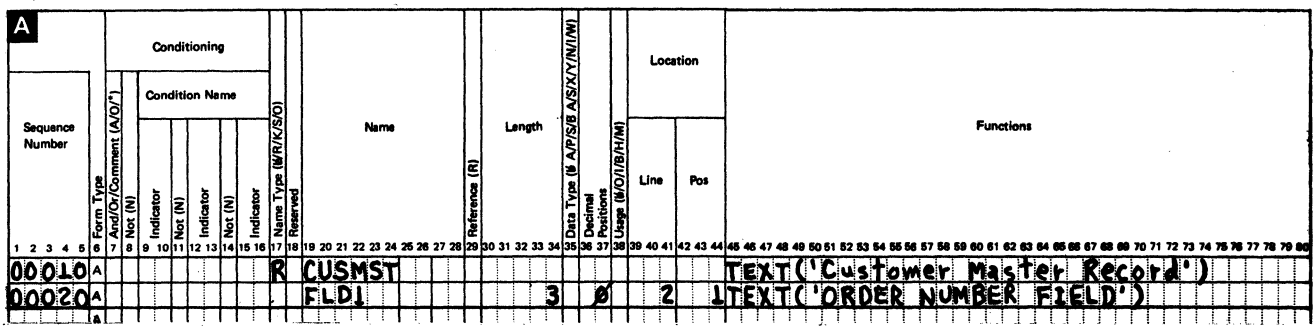
Use this record or field level keyword to supply a text description (or comment) for the record format or field. The text is used for program documentation.

If the length of the text is greater than 50 positions, only the first 50 characters are used by the high-level language compiler.

The TEXT keyword is valid for any record format or named field.

Option indicators are not valid with this keyword.

The following example shows how to specify the TEXT keyword at the record and field levels:



TIME

Use this field level keyword to specify that CPF is to retrieve the current system time and display it in a constant field. The time is retrieved each time your program issues an output operation. The following information is implied:

- The output-only attribute
- Numeric-only data type
- The edit word '0b:b b:b b' (If no editing is specified, this edit word is assumed.)
- A display length of 8 positions

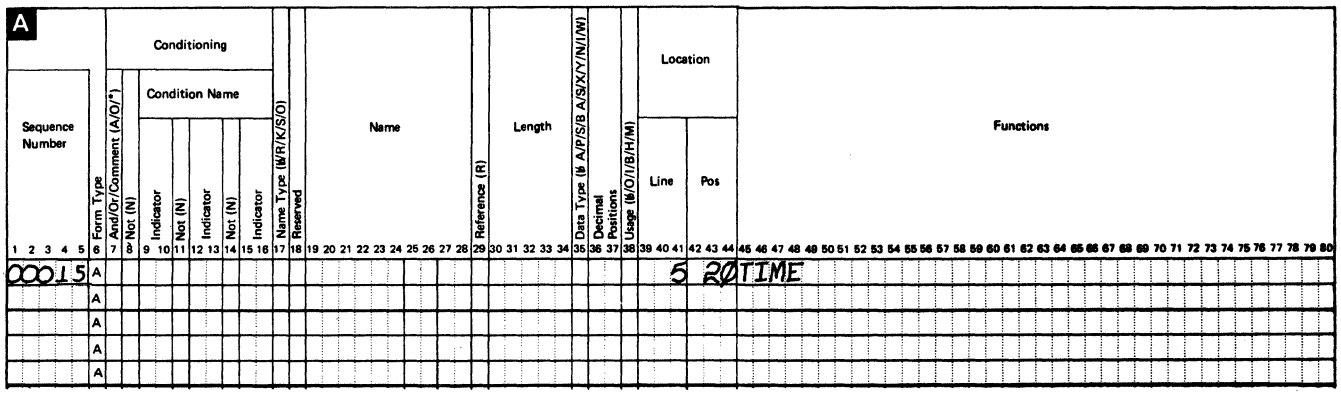
You can specify only the location of the field on the screen. Positions 17 through 38 must be blank.

You can also specify a different edit word (for example, EDTWRD('0bHRS:b bMINS:b bSECS.')) or one of the user-defined edit codes to be used in place of the implied edit word (see EDTCDE and EDTWRD keywords).

Option indicators are not valid with this keyword.

The TIME keyword is not valid for subfile records.

The following example shows how to specify the TIME keyword:



UNLOCK[(*ERASE) | (*MDTOFF)] | [(*ERASE *MDTOFF)] | [(*MDTOFF *ERASE)]

Use this record level keyword to specify that CPF is to unlock the keyboard immediately after issuing an input operation to the record format you are defining.

Without the UNLOCK keyword, CPF leaves the keyboard locked after reading the data on the display. The work station user cannot key data into input-capable fields while the data that has just been read is being processed.

The parameter values *ERASE and *MDTOFF are optional. If you do not specify any parameter value, *ERASE is the default.

When your program issues an input operation, the following sequence of operations usually occurs. The keyboard is unlocked (if it is not already unlocked) to allow the work station user to key into input-capable fields on the display. Then the work station user presses the Enter/Rec Adv key (or a valid command key). Input-capable fields in the record format have their modified data tags (MDTs) set on if they have been keyed into or if they were displayed with the DSPATR(MDT) keyword in effect. When the input operation is completed, the parameter values on the UNLOCK keyword affect the input-capable fields with MDTs set on as follows.

*UNLOCK (without GETRETAIN) or UNLOCK(*ERASE)*: The keyboard remains unlocked, input-capable fields on the display are erased, and their MDTs are not reset following the input operation.

*UNLOCK(*MDTOFF) or UNLOCK (with GETRETAIN)*: The keyboard remains unlocked, input-capable fields on the display are not erased, and their MDTs are reset following the input operation.

*UNLOCK(*ERASE *MDTOFF) or UNLOCK(*MDTOFF *ERASE)*: The keyboard remains unlocked, input-capable fields on the display with their MDTs set on are erased, and their MDTs are reset following the input operation.

The GETRETAIN keyword is ignored and an error message results at file creation time if GETRETAIN is specified with UNLOCK(any parameter).

Note: This keyword does not prevent your program from issuing an output operation immediately after an input operation. However, the keyboard is unlocked and the work station user could be keying input data when the output operation changes the display.

Option indicators are not valid with this keyword.

The following example shows how to specify the UNLOCK keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (b A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (b/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		RECORD1								UNLOCK(*ERASE)
00020	A										FLD1		4			B	2	2	
00030	A										FLD2		4			B	3	2	
	A																		
	A																		
00040	A								R		RECORD2								UNLOCK(*MDTOFF)
00050	A										FLD21		4			B	4	2	
00060	A										FLD22		4			B	5	2	
	A																		
	A																		
00070	A								R		RECORD3								UNLOCK(*ERASE *MDTOFF)
00080	A										FLD31		4			B	6	2	
00090	A										FLD32		4			B	7	2	
	A																		
	A																		
	A																		

USRDFN

Use this record level keyword to specify that the data for this record is in the form of a user-defined data stream. No fields are valid for this record because the data stream formats the screen. No file, record, or field level keywords apply to this record except PASSRCD, KEEP, PRINT, OPENPRT, TEXT, and the routing keywords. Once you do an output operation for this record, CPF no longer retains knowledge of the status of the records on the device. You should have in-depth knowledge of the device before using this keyword.

Option indicators are not valid with this keyword.

For more information on how to use this function, see the *CPF Programmer's Guide.*

The following example shows how to specify the USRDFN keyword:

A

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
	A																		
00020	A								R		USRREC								USRDFN
00030	A																		RTGPOS(1 2Ø 4Ø)
	A																		
	A																		
	A																		
	A																		
	A																		

VALUES(value-1 [value-2... [value-20]])

Use this field level keyword to specify a list of values that are valid for the user to enter into the field; CPF performs an implied equals test on the data entered against the value(s) you specify here.

The following example shows how to specify the VALUES keyword for a character and numeric values test:

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B/A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A										RESPC		11			I	8	2	VALUES('A' 'B' 'C' 'D')
00020	A										RESPN		31		Ø	I	9	2	VALUES(33 -42 Ø1)
00030	A										DECFLD		5		2	I	11	2	VALUES(1.2 100)
	A																		
	A																		
	A																		

There can be 1 to 20 values; specify them as parameter values with the keyword and separate them by at least one blank.

A value can be a numeric or a character literal, corresponding in length to the field that is to be tested. A character literal must be enclosed in apostrophes; a numeric literal is restricted to the digits 0 through 9 and can be preceded by a minus sign (-) for negative values. Alignment is on the low-order character position.

The compare is based on the value as it is passed to your program (for example, right-adjusted and padded, or left-adjusted and padded).

Option indicators are not valid with this keyword.

Note: If the value specified is a constant and the field named (to be compared with) is a numeric field, alignment is based on the decimal precision specified, and the leading and trailing blanks are filled with zeros. If no decimal point is specified, the decimal point is assumed to be to the right of the last (rightmost) digit. For example, for a numeric field with a length of 5 (specified in column 34) and 2 decimal positions (specified in column 37), 1.2 is interpreted as 001.20, and 100 is interpreted as 100.00.

VLDCMDKEY(response-indicator ['text'])

Use this file or record level keyword to specify that CPF is to set on the specified response indicator when any valid command key other than the Enter/Rec Adv key is pressed by the work station user.

One use of this function is to perform a simple test to determine if the work station user has requested a function you want to watch for in your program.

For a command key to be considered valid, you must have enabled the key by specifying it with one of the following keywords:

Keyword	Comments
CAnn	With or without response indicator
CFnn	With or without response indicator
CLEAR	With or without response indicator
HELP	With or without response indicator
HOME	With or without response indicator
PRINT	Only with a response indicator
ROLLUP	With or without response indicator
ROLLDOWN	With or without response indicator

The optional text is included on the listing generated at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The apostrophes are required. If you specify more than 50 characters between the apostrophes, the text is truncated to 50 characters on the program listing.

Option indicators are not valid with this keyword.

The following example shows how to specify the VLDCMDKEY keyword:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		REC1								VLDCMDKEY(90 'Any valid key')
00020	A																		CA01(91)
00030	A																		CA02(92)
00040	A																		CA03(93)
00050	A																		CLEAR(94)
	A																		
	A																		
	A																		

In this example, indicator 90 is set on if any of four keys (01, 02, 03, or Clear) is pressed.

Chapter 5. Printer Files

This chapter is divided into two sections. The first, *Positional Entries*, provides rules and examples for filling in positions 1 through 44 of the Data Description Specifications form. To find information in this section, first determine what position on the form to use, then look up the section describing that position.

The second section of this chapter, *Keyword Entries*, provides rules and examples for specifying DDS keywords. The keywords are described in alphabetical order.

For guidance in choosing positional entries and keywords for printer files, see the *CPF Programmer's Guide*.

When you are describing a printer file, specify the entries in the following order:

1. File level entries (optional)
2. Record level entries
3. Field level entries

} Repeat these entries for each record format in the file.

You must specify at least one record format in the file and at least one field in each record format.

Note: The file name is specified through the Create Printer File (CRTPRTF) command, not through DDS.

Complete printer file examples can be found in Appendix F, *Examples*.

An explanation of *file level*, *record level*, and *field level* can be found in Chapter 1, *Introduction*.

The syntax rules for specifying DDS keywords can be found in Chapter 1, *Introduction*.

The maximum number of record formats in a printer file is 1024. The maximum number of fields in any one record format is 32 767. The maximum combined length of all named fields and indicators in a record format is 32 767.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050*
Printed in U.S.A.

File | Programmer | Date | Keying Instruction | Graphic | Key | Description | Page of

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (M/R/I/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/I/O/I/B/H/M)	Location: Line	Pos	Functions
00010	A	*		PRINTER					F		ILE CODING	E	XAMPLE						
00020	A	*																	
00030	A								R		TITLER								SKIPB(3)
00040	A										FLD1		40					47	SPACEA(2) UNDERLINE
00050	A			50							FLD2		40					47	SPACEA(2) UNDERLINE
00060	A	*																	
00070	A								R		AUTHORR								
00080	A																	66	'by'
00090	A										FIELD1		40					47	SPACEB(1)
00100	A																		DFT('Task Force I')
00110	A			31															SPACEA(1)
00120	A			31														65	'and'
00130	A			31							FIELD2							47	
00140	A	*																	
00150	A								R		PUBR								SKIPB(58)
00160	A																	47	'Published by Department'
00170	A										DEPT		3		0			+1	
00180	A																	47	DATE EDTCDE(Y)
00190	A																		SPACEB(1)
00200	A		N	15															
00210	A	0		32		33		34										47	TIME
00220	A																		SPACEB(1)
	A																		

Figure 5-1. Printer File Coding Example

Positional Entries (Positions 1-44)

This section describes how to specify the first 44 positions of the Data Description Specifications form for printer files. To code the rest of the form, see the section *Keyword Entries (Functions Field, Positions 45-80)* later in this chapter.

Figure 5-1 shows some positional entries for printer files.

Positions 1-5 (Sequence Number)

Use these positions to specify a sequence number for each line on the form. The sequence number is optional and is used for documentation purposes only.

Position 6 (Form Type)

The A in this position designates this as a Data Description Specifications form. The form type is optional and is for documentation purposes only.

Position 7 (Comment)

An asterisk (*) in position 7 identifies the line as a comment line. Comment lines can appear anywhere in DDS and are retained only in the source file and printed on the source listing. (Comments are not printed on the expanded source listing.) Use positions 8 through 80 for comment text. A blank line (no characters specified in positions 7 through 80) is treated as a comment line.

Positions 7-16 (Conditioning)

Positions 7 through 16 are a multiple-field area in which you can specify option indicators. Option indicators are two-digit numbers from 01 to 99. Your program can set option indicators on (hex F1) or off (hex F0) to select a field or keyword.

A condition is an ANDed grouping of one to nine indicators that must all be in effect (set off if N is specified; set on if N is not specified) before the field or the keyword is selected. You can specify a maximum of nine indicators for each condition and nine conditions for each field or keyword; therefore, a maximum of 81 indicators can be specified for each keyword, when nine indicators are used with nine conditions.

When you specify a condition so that more than one indicator must be on or off before the condition is satisfied, it is called an AND condition; you are essentially saying that the first indicator you specified, AND the second, AND the third, and so on, must all be in effect before the condition is satisfied and the field or the keyword is selected. The field or the keyword must be specified on the same line as the last (or only) set of indicators specified.

You can also specify several conditions for the field or the keyword so that if any one of them is satisfied, the field or the keyword is selected. This is called an OR relationship, and it means that if the first condition is satisfied, OR the second condition, OR the third condition, and so on, the field or the keyword is to be selected. Note that conditions within the OR relationship can consist of just one indicator or can consist of several indicators ANDed together. Indicators can be ANDed to form a condition; conditions can be ORed to give your program several ways to select the field or the keyword.

AND (Position 7): If you need more than three indicators to form an ANDed condition, specify them in the same positions on the next line or lines. You can specify A in position 7 on the second or following lines to designate the continuation of the AND condition, or you can leave it blank because A is the default.

OR (Position 7): If you specify several conditions that are to be ORed together, each condition must start on a new line and each condition except the first must have an O in position 7. An O specified for the first condition produces a warning message, and that position is assumed to be blank.

NOT (Positions 8, 11, 14): If you want an indicator to be off instead of on to satisfy a condition, specify N in the position just preceding the indicator (position 8, 11, or 14).

Conditioning More Than One Field or Keyword: If you are conditioning a field, the field name (or the constant) and the last (or only) indicator must be on the same line. If you specify one or more keywords for the field, only the field is conditioned, not the keyword(s). If the field is not selected for an output operation, no keywords specified for that field are in effect, regardless of how the keywords are conditioned. For example, in Figure 5-1, if indicator 30 is off, SPACEA and UNDERLINE are not in effect.

If you want to condition one or more keywords, specify the last (or only) indicator on the same line as the keyword(s) and specify the keywords on the same line. If the conditioning applies to keywords on more than one line, keyword continuation must be used for the indicators to apply to all the keywords. See *Data Description Specifications Syntax Rules* in Chapter 1, *Introduction.*

Position 17 (Name Type)

The value in this position identifies the type of name in positions 19 through 28 (name). The valid entries for printer files are:

Entry	Meaning
R	Record format name
Blank	Field name

Figure 5-1 shows how to code the name type.

Position 18 (Reserved)

This position does not apply to any file and must be blank unless an asterisk appears in position 7 (indicates a comment line).

Positions 19-28 (Name)

Use these positions to specify record format names and field names. The following rules apply:

- Names must be 10 characters or less.
- Names must start in position 19.
- Names must begin with an alphabetic character (A through Z, @, $, and #). All subsequent characters can be alphameric (A through Z, 0 through 9, @, $, #, and underscore (_)). There can be no embedded blanks.

Figure 5-1 shows how to code the name field.

High-level languages can impose specific length and value restrictions on names. It is your responsibility to make sure that the names used here are acceptable to all language processors that process the file.

Record Format Name: When you specify R in position 17 (name type), the name specified in positions 19 through 28 is a record format name. You can specify more than one record format for a printer file, but each record format name must be unique within that file. You must specify field names or constant fields to complete a record format in a printer file.

Field Name: When you specify a blank in position 17 (name type), the name specified in positions 19 through 28 is a field name. Field names must be unique within the record format.

Constant Fields: Constant fields are unnamed fields (positions 19 through 28 must be blank). Further, the following rules apply to constant fields:

- Positions 17 through 38 must be blank.
- The location of the field is required (positions 39 through 44).
- The field can be conditioned using option indicators (positions 7 through 16).
- The constant itself is defined using one of the following entries:
  - Explicit DFT keyword (specify the literal within apostrophes with the DFT keyword)
  - Implicit DFT keyword (specify the literal within apostrophes without the DFT keyword)
  - DATE keyword (specify no literal; see the DATE keyword description)
  - TIME keyword (specify no literal; see the TIME keyword description)
  - PAGNBR keyword (specify no literal; see the PAGNBR keyword description)
- The EDTCDE or the EDTWRD keyword can be specified only when DATE, TIME, or PAGNBR is also specified.

You must specify the fields in the order in which they are to appear on the printed page unless you specify line numbers. If you use line numbers, you can specify the fields in any order.

Position 29 (Reference)

Specify R in this position to use the reference function of CPF to copy the attributes of a previously defined named field (called the *referenced field*) to the field you are defining now. The referenced field can be previously defined in the printer file you are defining; or it can be in a previously created data base file (the data base file to be referenced is specified on the REF or REFFLD keywords). The field attributes referenced are the length, the data type, and the decimal positions of the field, as well as editing and text keywords.

If you do not specify R, you cannot use the reference function for this field and you must specify field attributes for this field.

Position 29 must be blank at the file and record levels.

The name of the referenced field can be either the same as the field you are defining or different from the field you are defining. If it is the same, you need only specify R in position 29 in addition to specifying the name of the field you are defining in positions 19 through 28. If it is different, you must specify the name of the referenced field with the REFFLD keyword.

You can specify the name of the file defining the referenced field as a parameter value with the REF or the REFFLD keyword. See REF and REFFLD keyword descriptions later in this chapter and Appendix A, *How to Specify REF and REFFLD*, for explanations of how CPF finds the referenced field.

You do not need to copy all attributes from the previously defined field to the field you are defining. To override specific attributes of the referenced field, specify those attributes for the field you are defining, as follows:

- To override editing keywords (EDTCDE or EDTWRD), specify EDTCDE or EDTWRD for the field you are defining. (You can simply delete these keywords by specifying DLTEDT for the field you are defining.)

- Validity checking keywords (CHECK, COMP, RANGE, VALUES), if specified for the referenced field, are ignored in the printer file.

When you override some specifications, others are also affected, as follows:

- If you specify a value for data type, field length, or decimal positions for the field you are defining, editing keywords are not copied from the referenced field.

- Packed decimal and binary fields are not supported for printer files; therefore, when you reference fields of these types, the data type is converted to zoned decimal (S in position 35) in the printer file.

Note: Once the printer file is created, the referenced file can be deleted or changed without affecting the field definitions in the printer file. To incorporate changes made in the referenced file, delete and re-create the printer file.

Positions 30-34 (Field Length)

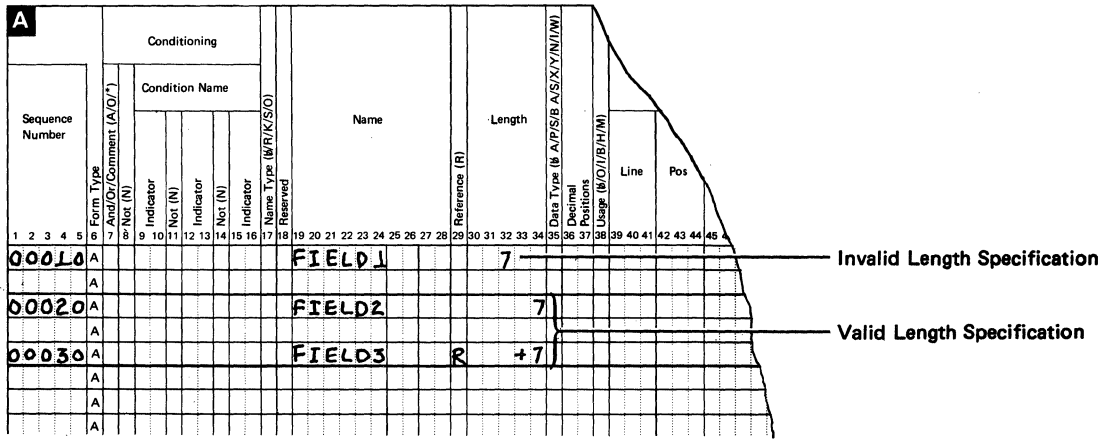
You must specify a field length for each named field unless it is being duplicated from a referenced field. Your entry here represents the number of bytes of data to be passed from your program when an output operation is done for this field. (If the field is to be edited, the associated edit code or edit word is used to determine the printed length of the field.) Figure 5-1 shows how to code the field length.

The maximum length of a zoned decimal field is 31. Data description specifications allow a maximum field length of 32 767 characters. If the field length causes the field to extend beyond the form size, a warning diagnostic is issued.

When you specify this field, the entry must be right-justified; leading zeros are optional.

If you are using a referenced field, you can override the length of the field by specifying a new value or by specifying the increase or decrease in length. To increase the length, specify +n, where n is the increase. To decrease the length, specify -n, where n is the decrease. For example, an entry of +4 indicates that the field is to be 4 digits longer than the referenced field. The field length can be overridden without overriding the decimals.

The following example shows valid and invalid field length specifications:



Position 35 (Data Type)

Specify in this position the data type associated with the field. The valid entries for this field for a printer file are:

Entry	Meaning
S	Zoned decimal
A	Character

If you do not specify a data type and do not duplicate one from a referenced field, CPF assigns a default value as follows:

- A (character) if the decimal positions (36 through 37) are blank
- S (zoned decimal) if the decimal positions (36 through 37) contain a number in the range 0 through 31

Note: Placing 0 in position 37 is a convenient way to specify an integer numeric field.

Figure 5-1 shows how to code the data type.

Positions 36-37 (Decimal Positions)

Use these positions to specify the decimal placement within a zoned decimal field and also to specify the data type of the field as it appears in your program.

If you leave these positions blank, CPF assumes a data type of character for the field.

If you enter a number in these positions, CPF assumes a data type of zoned decimal for the field. The number specified is the number of positions to the right of the decimal point; it must be less than or equal to the field length, with a maximum of 31. Figure 5-1 shows how to code decimal positions.

If you are using a referenced field, you need not specify decimal positions. The information is retrieved from the referenced file. The decimals retrieved can be either overridden or changed. To override the decimal positions, specify the new value explicitly. To change the decimal positions, specify the amount you want the field increased or decreased and precede it with either + or -, respectively. For example, an entry of +4 indicates there are four more digits to the right of the decimal point than in the referenced field.

Position 38 (Usage)

The only valid entries for this field are either blank or 0, and either entry means output only.

Positions 39-44 (Location)

Use these positions to specify the exact location on the page where the beginning of the field appears. The following conditions apply:

- When line numbers are specified, fields can appear in any order. They will be resequenced into line-position order when placed in the printer file.
- When line numbers are not specified, the field ordering within the printer file is the same as specified in the DDS.
- When fields or space/skip keywords are conditioned, they are treated as if they were selected when performing overlap checks.
- In a record format having several fields, when a field that has skip/space keywords is conditioned or a field-level skip/space keyword is conditioned, a warning message is issued indicating that overlapping fields could occur and not be diagnosed.
- When the form size is exceeded because of a field's length, location, or associated skip/space keywords, or a combination of these, a warning message is issued.
- The maximum value for line number is 255. The maximum value for position number depends on the characters-per-inch (CPI) parameter on the Create Printer File (CRTPRTF) command. If that value is 10, which is the default, the maximum is 132; if it is 15, the maximum is 198.
- The overflow line (the last printed line on a page) depends on the OVRFLW and FORMSIZE parameters on the Create Printer File (CRTPRTF), Change Printer File (CHGPRTF), and Override with Printer File (OVRPRTF) commands. For externally described files, RPG cannot control page overflow. See the *CPF Programmer's Guide* for input/output considerations regarding the overflow line.
- When externally described files are used by high-level language compilers, the fields are sequenced in the output record area according to the DDS. Refer to the appropriate high-level language manual for specific information. If fields overlap, the printer overprints. See the expanded source in the compiler listing generated by the Create Printer File (CRTPRTF) command for field lengths and output buffer positions.

Positions 39-41 (Line): The entry you make in these positions specifies the line on the page in which the field begins. The entry must be right-justified; leading zeros are optional. Line numbers can be specified for either named or unnamed fields within a record; if you specify a line number for one field within a record, you must specify either a line number in positions 39 through 41 or an increment (+n) in positions 42 through 44 for *all* fields within that record.

Line numbers are not valid if one of the skip or space keywords has been specified at either the record level or field level. Line numbers are valid, however, if one of the skip keywords has been specified at the file level. (Space keywords are not valid at the file level.)

Positions 42-44 (Position): The entry you make here specifies the starting position of the field. It must be right-justified; leading zeros are optional.

If you specify a location of a field in a record and the field is not ignored, you can specify the location of subsequent fields within that record by leaving the line number blank and specifying an increment (+n) for 42 through 44 (position entry). The increment indicates the number of spaces to be left between the end of the previous field and the beginning of the field you are defining. If you specify an increment, the line number entry must be blank. Note that if the increment causes an implicit space operation, and line numbers are not being used for the record format, then skip/space keywords must be used to cause spacing to occur. The following example illustrates this problem and two possible solutions (for a form width of 132 characters):

A

```
     A*
     A* POSITION INCREMENT CAUSES PRFL01 TO OVERLAP PRFL02
     A*
     A          R PRTOUT                    SKIPB(1)
     A            PRFL01        130          1TEXT('START LOC 1,1 END LOC 1,130')
     A            PRFL02        130         +2TEXT('OVERLAPS PRFL01')
     A*
     A* SOLUTION 1 TO PREVENT OVERLAP IS TO SPECIFY SPACEA OR SKIPA WITH PRFL01
     A*   OR TO SPECIFY SPACEB OR SKIPB WITH PRFL02
     A*
     A          R PRTOUT2                   SKIPB(1)
     A            PRFL01A       130          1
     A            PRFL02A       130         +2SPACEB(1)
     A*
     A* SOLUTION 2 PROVIDES A FUNCTIONAL EQUIVALENT FORMAT NOT USING SKIP/SPACE
     A*
     A          R PRTOUT3
     A            PRFL01B       130       1  1
     A            PRFL02B       130         +2
     A
```

Any field that extends beyond the end of a line is continued on the next line. The break occurs at the end of the line but you can cause it to be folded at a blank by specifying the BLKFOLD keyword. This assumes that FOLD(*YES) is in effect for the file (see CRTPRTF, CHGPRTF, and OVRPRTF commands in the *CL Reference Manual*).

The data description specifications determine which fields appear on the same print line. The data description processor diagnoses overlap at file creation time if it occurs. Keywords or fields containing keywords that are optioned are assumed to be selected. Therefore, no overlap check is made for the cases where the keywords or fields are not selected. In the following example, no fields would overlap unless indicator 01 is on, in which case F1, F3, and F4 would overlap. A diagnostic is issued for the format indicating that field selection or conditioning of space/skip keywords can cause fields to overlap at execution time.

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	A	*																	
	A	*		OVERLAPPING FIELDS ONLY IF IND 01 IS ON															
	A	*																	
	A								R		REC1								SKIPB(1)
	A										F1		1					1	
	A		N01								F2		1					1	SPACEB(1) SPACEA(1)
	A										F3		1					1	
	A										F4		1					1	
	A		N01																SPACEB(1)
	A																		
	A																		

Keyword Entries (Positions 45-80)

This section contains keyword entries valid for defining printer files. They are entered in positions 45 through 80 (functions).

See the section on *Data Description Specifications Syntax Rules* in Chapter 1 for a discussion of the general rules for specifying keywords. Figure 5-2 shows how to use the general rules to specify DDS for printer files:

IBM International Business Machines Corporation — **DATA DESCRIPTION SPECIFICATIONS** — GX21-7754-1 UM/050* Printed in U.S.A.

File | Keying Instruction | Graphic / Key | Description | Page of
Programmer | Date

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Pos	Functions
00010	A	*		SYNTAX FOR A PRINTER FILE															
00020	A	*																	
00030	A																		KEYWORDA
00040	A								R		RECORDA								KEYWORDB
00050	A																		KEYWORDC KEYWORDD
00060	A										FIELDA		20				1	3	KEYWORDE('This is a text example')
00070	A																		KEYWORDF(VALUEA)
00080	A																		KEYWORDG(VALUEB VALUEC)
00090	A										FIELDB		40				2	3	
00100	A										FIELDC		5	2		3		3	KEYWORDH('This text example continues with a minus sign')
00110	A																		
00120	A																		KEYWORDI('This text example +
00130	A																		continues with a plus sign')
00140	A																4	3	'This literal implies the DFT +
00150	A																		keyword and an unnamed field +
00160	A																		starting at line 4, position 3'
	A																		
	A																		

You can specify option indicators in the boxed-in positions.

Figure 5-2. Syntax for a Printer File

The following keywords are valid for printer files:

- BLKFOLD
- DATE
- DFT
- DLTEDT
- EDTCDE
- EDTWRD
- INDTXT
- PAGNBR
- REF
- REFFLD
- SKIPA
- SKIPB
- SPACEA
- SPACEB
- TEXT
- TIME
- UNDERLINE

BLKFOLD

Use this field level keyword (blank fold) for named fields that overflow onto subsequent print lines, to cause folding to occur at a blank rather than at the end of a line. The default is to fold at the end of the physical print line.

When you use this keyword, the field length is not increased; therefore, a portion of the output data might be truncated.

Option indicators are not valid with this keyword although option indicators can be used to condition the field (whether named or constant) with which it is associated.

The following example shows how to specify the BLKFOLD keyword:

A

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00010	A										FIELD1		18Ø	A			1Ø	2Ø	BLKFOLD
	A																		
	A																		

DATE

Use this field level keyword to specify that CPF is to retrieve the current job date (the date the job was started) and print it in a constant field. The following information is implied for the field and you cannot specify it:

- The output-only attribute
- A field length of 6 positions

You must specify only the location of the field on the page. Positions 17 through 38 must be blank.

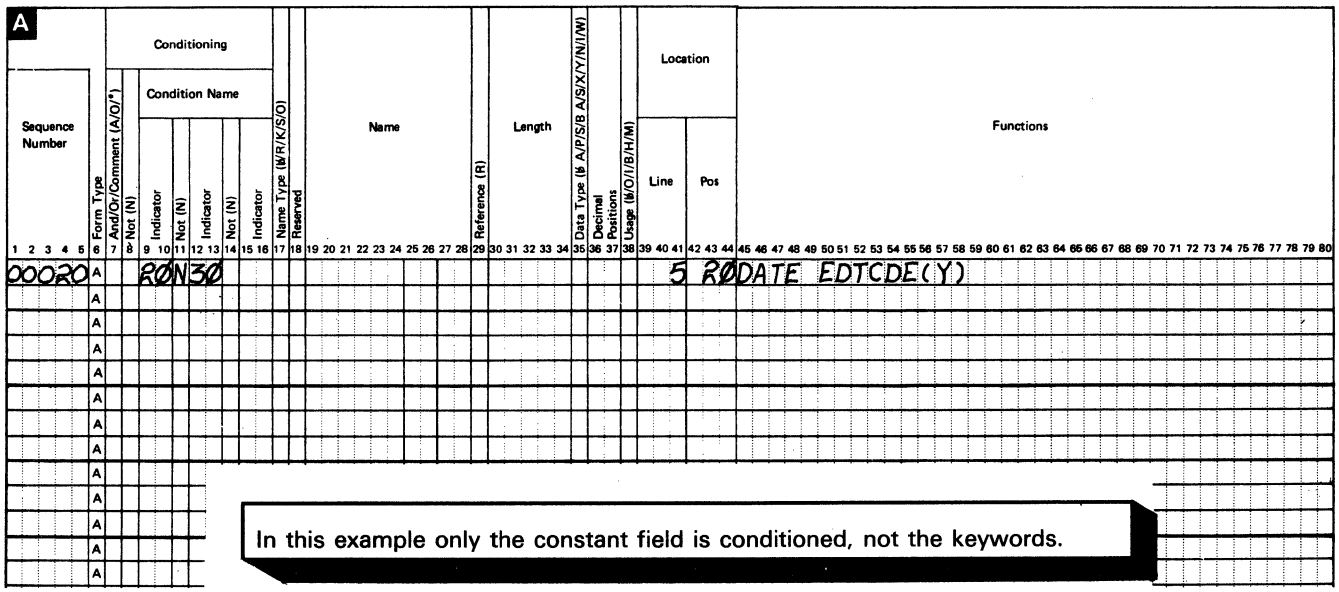
You can optionally specify an edit code or an edit word for this field (see the EDTCDE and EDTWRD keywords). If no edit code or edit word keyword is specified, no editing takes place.

Option indicators are not valid for this keyword.

Notes:

1. The Y edit code uses the character in the system value QDATSEP, as the date separator character. When DATE and EDTCDE(Y) are specified together on a constant (unnamed) field, the data is edited according to the system values QDATSEP (date separator character) and QDATFMT (date format). See the description of the EDTCDE keyword for more information on how the date is edited.
2. At execution time, if QDATFMT is JUL (Julian), and any edit code is specified, CPF puts a zero in the sixth position. EDTCDE(Y) should be used.

The following example shows how to specify the DATE keyword:



DFT('literal') | ('literal')

Use this field level keyword to specify the constant value for constant fields (unnamed fields). The specified literal must be enclosed in apostrophes.

If you omit the DFT keyword, specifying the literal by itself (still within apostrophes) implies the DFT keyword. Specifying the DFT keyword implicitly or explicitly for constant fields provides the same function.

When you specify the DFT keyword, whether implicitly or explicitly, the number of characters between apostrophes is the length of the constant field (except when you specify double apostrophes within the constant; the double apostrophes appear as one apostrophe).

Option indicators are not valid with this keyword, although option indicators can be used to condition the constant field with which it is specified.

The following example shows how to specify the DFT keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		SUPPLIES								
00020	A										PENS		20				2		DFT('Plastics')
00030	A										INK		20				3		DFT('Blue')
00040	A										PAPER		20				4		DFT('White')
00050	A																7		DFT('ON')
00060	A																8		'ON'
00070	A																		
00080	A			01													12		'Hotel name: ''Terrace Inn'''
00100	A																		
00110	A			02													12		'Hotel name: ''RiverView Inn'''
	A																		
	A																		
	A																		

The specifications DFT('ON') and 'ON' are equivalent and show the difference between specifying the DFT keyword explicityly and implicity.

If indicator 01 is on, the following is printed:
Hotel name: 'Terrace Inn'
If indicator 02 is on and indicator 01 is off, the following is printed:
Hotel name: 'Riverview Inn'

DLTEDT

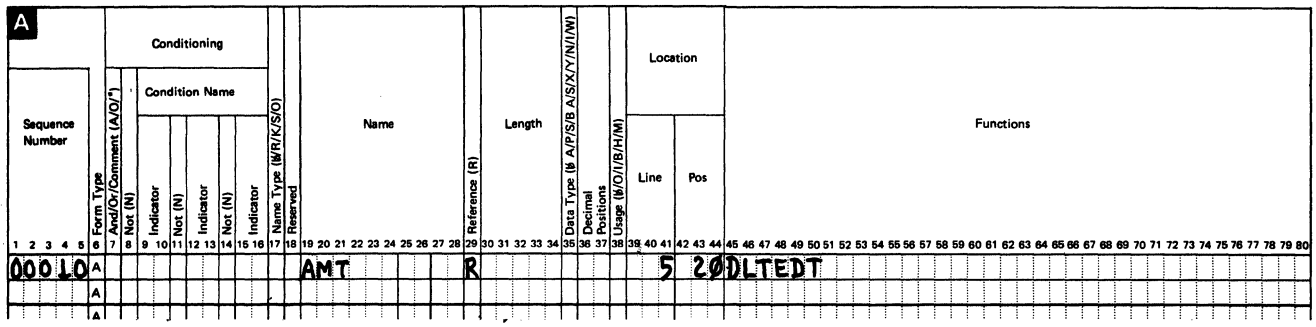
If a field description is duplicated from a data base file, this field level keyword prevents certain information from being duplicated. Use it to specify that CPF is to ignore any edit code or edit word keywords that were specified for the referenced field.

This keyword is valid only when you have specified R in position 29 for this field, and have also specified either the REF or the REFFLD keyword.

If replacement edit information is needed, simply specify the EDTCDE or the EDTWRD keyword for the field you are defining; the new keyword overrides the editing from the referenced field. In this case you do not need to specify the DLTEDT keyword.

Option indicators are not valid with this keyword.

The following example shows how to specify the DLTEDT keyword:



EDTCDE(edit-code [* | floating-currency-symbol])

Use this field level keyword to specify an edit code for editing of output data for a numeric field using this current field description. An edit code is a required parameter for this keyword.

EDTCDE is valid only for numeric fields. Both EDTCDE and EDTWRD keywords cannot be specified for the same field. You can use the EDTCDE keyword to perform edit functions such as the following:

- Suppress leading zeros
- Omit a sign from the low-order position of the field
- Punctuate the field without having to specify an edit word (see the EDTWRD keyword)
- Perform asterisk fill
- Specify floating currency symbol (this symbol must correspond to the system value QCURSYM)
- Perform user-defined edit functions

Option indicators are not valid with this keyword.

CPF Edit Codes: The edit codes that you can specify as a parameter value for this keyword are:

1 through 4
A through D
J through M
Y and Z

Edit code X has no effect in printer files. For an explanation of its use, see the EDTCDE keyword descriptions in Chapter 2, *Physical Files* and Chapter 3, *Logical Files*.

The following chart summarizes the functions provided by CPF edit codes. A negative field can be punctuated with no sign, CR (credit), or a minus.

Edit Code	Commas[1] Printed	Decimal Points[1] Printed	Signs Printed When Negative Number			Zero Balance Printed	Leading Zero Suppressed
			No Sign	CR	- (Minus)		
1	Yes	Yes	No sign			Yes	Yes
2	Yes	Yes	No sign				Yes
3		Yes	No sign			Yes	Yes
4		Yes	No sign				Yes
A	Yes	Yes		CR		Yes	Yes
B	Yes	Yes		CR			Yes
C		Yes		CR		Yes	Yes
D		Yes		CR			Yes
J	Yes	Yes			–	Yes	Yes
K	Yes	Yes			–		Yes
L		Yes			–	Yes	Yes
M		Yes			–		Yes
Y[2]							Yes
Z[3]							Yes

[1]The QDECFMT system value determines the decimal point character (period in U.S. usage), the character used to separate groups of three digits (comma in U.S. usage), and the type of zero suppression (depending on comma and period placement). See the *CPF Programmer's Guide* for detailed instructions.

[2]The Y edit code suppresses the leftmost zero of a date field that is three to six digits long, and it suppresses the two leftmost zeros of a field that is seven positions long. The Y edit code also inserts slashes (/) between the month, day, and year according to the following pattern:

nn/n
nn/nn
nn/nn/n
nn/nn/nn
nnn/nn/nn

If the DATE keyword is specified with EDTCDE(Y), the separator character used is the system value, QDATSEP. The slash (/) is the default QDATSEP. If QDATFMT is JUL (Julian), the data is normally formatted as: nn/nnn.

[3]The Z edit code removes the sign (plus or minus) from a numeric field. The sign of the units position is changed to a hexadecimal F before the field is written.

User-Defined Edit Codes: You can also specify any of the five user-defined edit codes (5 through 9). Before creating a device file from DDS source containing user-defined edit codes, you must create and define the user-defined edit codes using the Create Edit Description (CRTEDTD) command. When you create the printer file, the editing information is extracted from the user-defined edit codes as referenced in the DDS. Changing user-defined edit codes does not affect existing device files unless the device files are re-created. See the *CPF Programmer's Guide* and the *CL Reference Manual* for a description of how to create user-defined edit codes.

Asterisk Fill and/or Floating Currency Symbol: You can optionally specify asterisk fill and/or floating currency symbol with edit codes 1 through 4, A through D, and J through M.

When you specify asterisk fill, an asterisk (*) is written for each zero suppressed. A complete field of asterisks is printed for a zero balance field.

When you specify floating currency symbol, the symbol appears to the left of the first significant digit. The symbol does not print on a zero balance when an edit code is used that suppresses the zero balance. The symbol that you specify must match the system value for the floating currency symbol. (The symbol must match when the file is created; it does not have to match when the file is used.)

Note: If an edit code is changed after a file is created, the editing specified at the time the file was created is used. The new edit code is not used unless the file was re-created.

Figure 5-3 shows valid edit codes with examples of unedited source data and edited output. The zero suppression and decimal characters are determined by the system value QDECFMT. The date separator character is determined by the system value QDATSEP. In this figure, QDECFMT is assumed to equal b, and QDATSEP is assumed to equal /.

Edit Codes	Positive Number– Two Decimal Positions	Positive Number– No Decimal Positions	Negative Number– Three Decimal Positions[1]	Negative Number– No Decimal Positions[1]	Zero Balance– Two Decimal Positions	Zero Balance– No Decimal Positions
Unedited	1234567	1234567	ƀƀƀ12ƀ	ƀƀƀ12ƀ	000000	000000
1	12,345.67	1,234,567	.120	120	.00	0
2	12,345.67	1,234,567	.120	120		
3	12345.67	1234567	.120	120	.00	0
4	12345.67	1234567	.120	120		
A	12,345.67	1,234,567	.120CR	120CR	.00	0
B	12,345.67	1,234,567	.120CR	120CR		
C	12345.67	1234567	.120CR	120CR	.00	0
D	12345.67	1234567	.120CR	120CR		
J	12,345.67	1,234,567	.120–	120–	.00	0
K	12,345.67	1,234,567	.120–	120–		
L	12345.67	1234567	.120–	120–	.00	0
M	12345.67	1234567	.120–	120–		
Y[2]	123/45/67	123/45/67	0/01/20	0/01/20	0/00/00	0/00/00
Z[3]	1234567	1234567	120	120		

[1]The ƀ represents a blank. This may occur if a negative zero does not correspond to a printable character.
[2]The Y edit code suppresses the leftmost zero of a date field that is three to six digits long, and it suppresses the two leftmost zeros of a field that is seven positions long. For more information, see the description of this edit code under *CPF Edit Codes* in this keyword description.
[3]The Z edit code removes the sign (plus or minus) from a numeric field and suppresses leading zeros of a numeric field.

Figure 5-3. Valid Edit Codes, Source Data, and Edited Output

The following example shows how to specify the EDTCDE keyword:

A

Sequence Number (1–5)	Form Type (6)	Conditioning (7–16)	Name Type (17)	Reserved (18)	Name (19–28)	Reference (29)	Length (30–34)	Data Type (35)	Decimal Positions (36–37)	Usage (38)	Location Line (39–41)	Location Pos (42–44)	Functions (45–80)
00050	A				PRICE		5		2		5	2	EDTCDE(J *)
	A												
	A												

EDTWRD('edit-word')

If you cannot accomplish the desired editing by using the EDTCDE keyword, you can specify an edit word instead. Make it up according to the rules listed below. The edit word specifies the form in which the field values are to be printed and clarifies the data by inserting characters directly, such as decimal points, commas, floating and fixed currency symbol, and credit balance indicators. It can also be used to suppress leading zeros and to provide asterisk fill protection.

Use the following rules to specify a valid edit word:

- Enclose the edit word in apostrophes.
- The EDTWRD keyword is valid only for numeric fields.
- The number of replaceable characters in the edit word must equal the length of the field.
- When using the floating currency symbol, the sum of the number of blanks and the stop-zero-suppression character (digit positions) contained in the edit word must be equal to the number of positions in the field to be edited. The currency symbol is not counted as a digit position. For example, if you specify a field length of 7 with 2 decimal positions, the edit word must be specified as

  EDTWRD('ƀƀƀƀ$0.ƀƀ')

  where the ƀ represents a blank.

- Any printable character is valid, but the following characters in certain positions have special meanings:

  - *Blank*: A blank is replaced with the character from the corresponding position of the data field. A blank position is referred to as a digit position.

  - *Ampersand*: An ampersand causes a blank in the edited field. The ampersand is not printed.

  - *Zero*: A zero stops zero suppression; place it in the rightmost position where zero suppression is to stop. The zero is replaced with the character from the corresponding position of the data field, unless that character is a zero. Any zeros in the data that appear to the right of the stop-zero-suppression character are printed. At least one leading zero is suppressed. Each leading zero that is suppressed is replaced by a blank.

  - *Asterisk*: An asterisk stops zero suppression and replaces leading zeros with asterisks (asterisk protection). Place the asterisk in the rightmost position where zero suppression is to stop. Each zero that is suppressed is replaced by an asterisk. An asterisk preceding a zero is interpreted as representing asterisk protection; in this case, the zero prints as a constant.

    An asterisk is considered a digit-replace position.

  - *Currency Symbol*: A currency symbol coded immediately to the left of the zero suppression code causes the insertion of a currency symbol in the position to the left of the first significant digit. It is called the *floating currency symbol* when used in this manner.

    A currency symbol coded in the leftmost position of the edit word is fixed and prints in the same location each time. When used in this manner it is called the *fixed currency symbol.*

    The currency symbol is not considered a digit replace position. This symbol must correspond to the system value QCURSYM.

  - *Decimals* and *Commas*: Decimals and commas are printed in the same relative positions in which they are coded in the edit word unless they are to the left of the first significant digit. In that case, they are blanked out or replaced by an asterisk.

  - *All Other Characters*: All other characters are printed if they are to the right of significant digits in the data field. If they are to the left of the high-order significant digits in the data word, they are blanked out or replaced by asterisks if asterisk protection is being used.

- If you want to show a negative sign with a negative number, include a sign in the edit word. Use either the minus sign (-) or the letters CR (credit). These print only if the number is negative.

- Both EDTWRD and EDTCDE cannot be specified for the same field.

Option indicators are not valid with this keyword.

The following example shows how to specify the EDTWRD keyword:

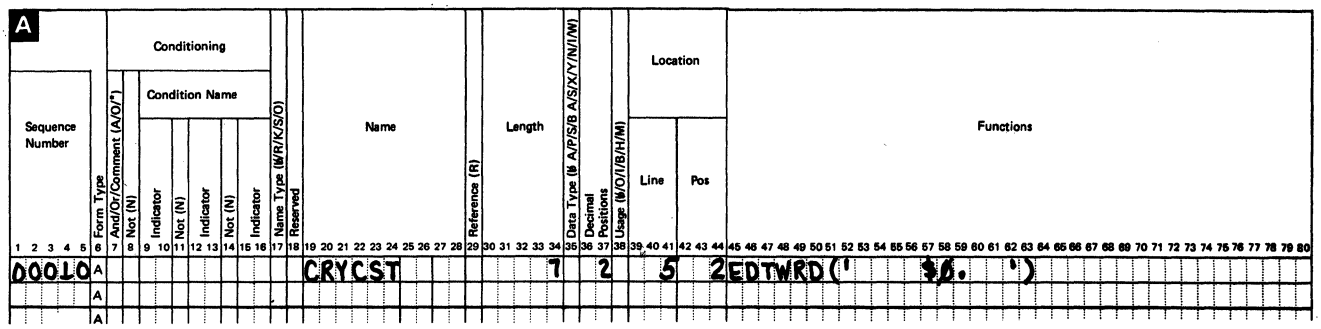


Figure 5-4 shows sample edit words with the program value of the field and the printed value of the field.

Edit Word	Program Value	Printed As
' , , 0. &CR*'	0000000005-	.05 CR*
' , ,$0 . -*'	0000000005+	$0.05 *
' , , $0. CR**'	0034567890-	$345,678.90CR**
'$, , 0. '	0000000000	$ .00
'$& , , 0 . &-&GROSS'	1234567890-	$ 12,345,678.90 - GROSS
' , , * . &-'	0000135792	*****1,357.92
	0000135792	0000135792
	0000135792-	0000135792
' '	0000000000	
' '	0000135678+	135678
' '	0000135678-	135678
' 0'	0000135678-	135678
'0 '	0000135678+	0000135678
'$ &-&NET'	0000135678+	$ 135678 NET
'$ &-&NET'	0000135678-	$ 135678 - NET
'$0 -&NET'	0000135678	$ 0000135678 NET
' $0 &CR*'	0000135678-	$135678 CR*
' $0 &CR*'	1234567809-	$12345678 09 CR*
' *&CR'	0000000000-	********** CR
' * &CR'	0000000000-	*********00 CR
'* '	0000135678-	*0001356 78
' , , . &CR*&NET'	0000135678-	1,356.78 CR* NET
' , , . &CR*&NET'	0000135678	1,356.78 * NET
' , , $0. '	0000000005	$.05
' , , $0. CR'	0001356789-	$13,567.89CR
' , , *. *CR**'	0000135678+	*****1,356.78* **
' , , DOLLARS CENTS'	0000135678	1,356DOLLARS78CENTS
'0 - - '	095140036	95-14-0036
'& ,* 0, '	013579	**130,579
' - - &LATER'	093076	9-30-76 LATER
' & & &LATER'	093076	9 30 76 LATER
' / / '	100176	10/01/76

Figure 5-4. Sample Edit Words

INDTXT(indicator 'indicator-usage-text')

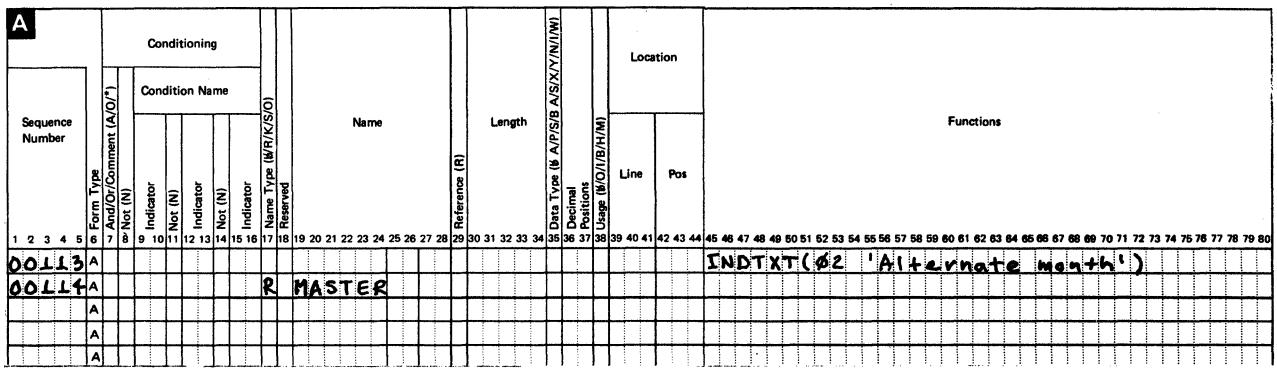
Use this file, record, or field level keyword to associate descriptive text (indicating intent or usage) with a specific option indicator. You can specify it once for each indicator.

If you specify this keyword, indicator usage text is a required parameter value. It must be a character literal and must be enclosed in apostrophes. If the length of the text is greater than 50 positions, only the first 50 characters are used by the high-level language compiler.

Option indicators are not valid with this keyword.

Note: This specification by itself does not cause the specified indicator to appear in the output record area. It merely provides text to be associated with the indicator. If the indicator has not been specified elsewhere (either as an option or a response indicator), then the text is lost without a diagnostic. Also, once an indicator has been given a textual assignment (either by this keyword or the response indicator text), no other textual assignment is made.

The following example shows how to specify the INDTXT keyword:



PAGNBR

Use this field level keyword to specify the location of an unnamed, four-digit, zoned decimal field to contain the page number. Specify only the PAGNBR keyword and the location of the field. (The location of the field can be either position only, or line number and position.)

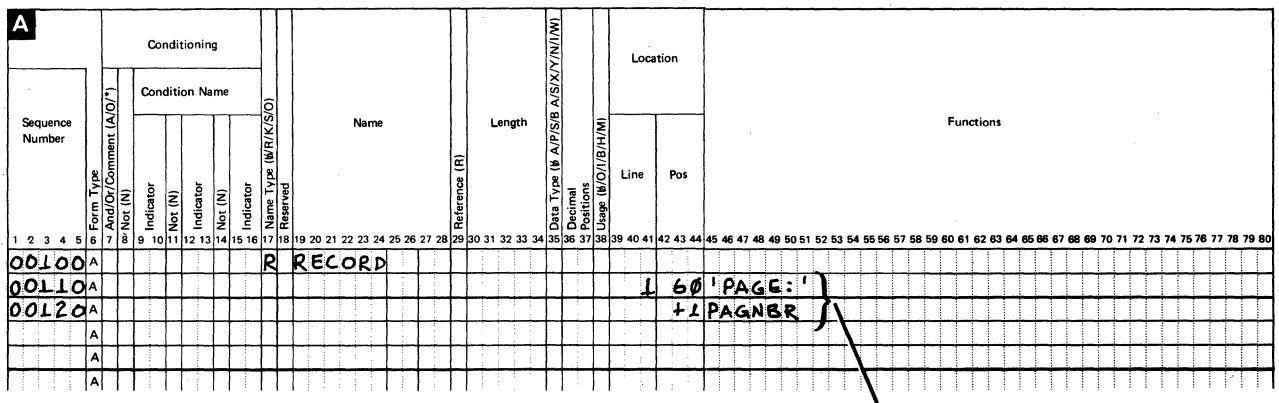
When the printer file is opened, CPF sets the page count to zero and increments it by one before printing each new page whether or not the PAGNBR keyword is specified. The page number is printed each time any field for which the PAGNBR keyword is specified is printed. The page number is not incremented beyond 9999; it stays 9999 until it is reset. To reset the page count, condition PAGNBR with option indicators; CPF resets the page number when your program selects PAGNBR. See *On Separate Lines with Separate Indicators* later in this keyword description.

You can optionally specify EDTCDE or EDTWRD with the PAGNBR keyword.

Option indicators are valid with this keyword.

The field (location only) and the keyword can be specified on one line with no indicators or on separate lines with separate indicators. The following sections explain the differences.

On One Line with No Indicators: The page number is always printed and you cannot reset it to one.



Two constant fields are specified: 'PAGE:' and the page number itself (location specified as +1).

On Separate Lines with Separate Indicators: If the field indicator (01 in the following example) is off, the field is not printed, no matter what the keyword indicator (02 in the following example) is set to. If the field indicator is on, the field is printed. The page count is incremented as long as the keyword indicator is not changed (the keyword indicator can be off or on and the page number is still printed). However, if the keyword indicator is changed from off to on, the page count is reset to one.

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/Ni)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12 13	14 15 16			17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00100	A								R		RECORD								
00110	A			01													1	60	'PAGE'
00120	A			01														+1	
00130	A			02															PAGNBR
	A																		
	A																		
	A																		

REF(data-base-file-name[.library-name][format-name])

Use this file level keyword to specify the name of a file from which field descriptions are to be retrieved. Use it when you want to duplicate descriptive information from one or more fields in a previously described record format. You can code the file name once here rather than on the REFFLD keyword with each of the field descriptions that reference the file.

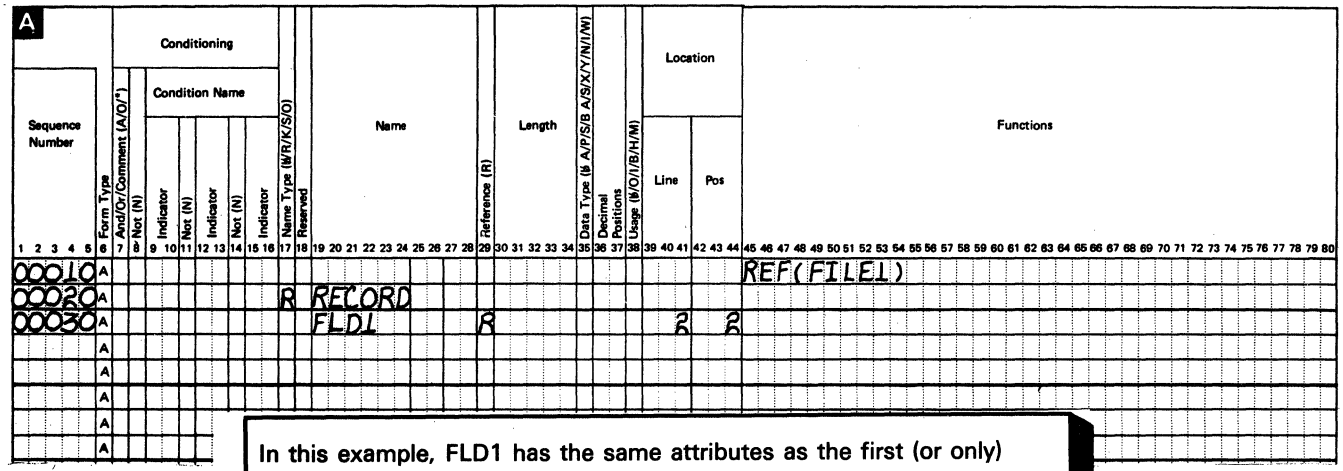
If there is more than one record format in the referenced file, specify a record format name as a parameter value for this keyword to tell CPF which one to use, unless the formats should be searched sequentially.

The file name is required for this keyword; the record format name and the library name are optional.

If you do not specify the library name, the current library list is used. If the record format name is not specified, each format is searched in order (as they are specified); the first occurrence of the field name is used. For more information, see Appendix A, *How to Specify REF and REFFLD*.

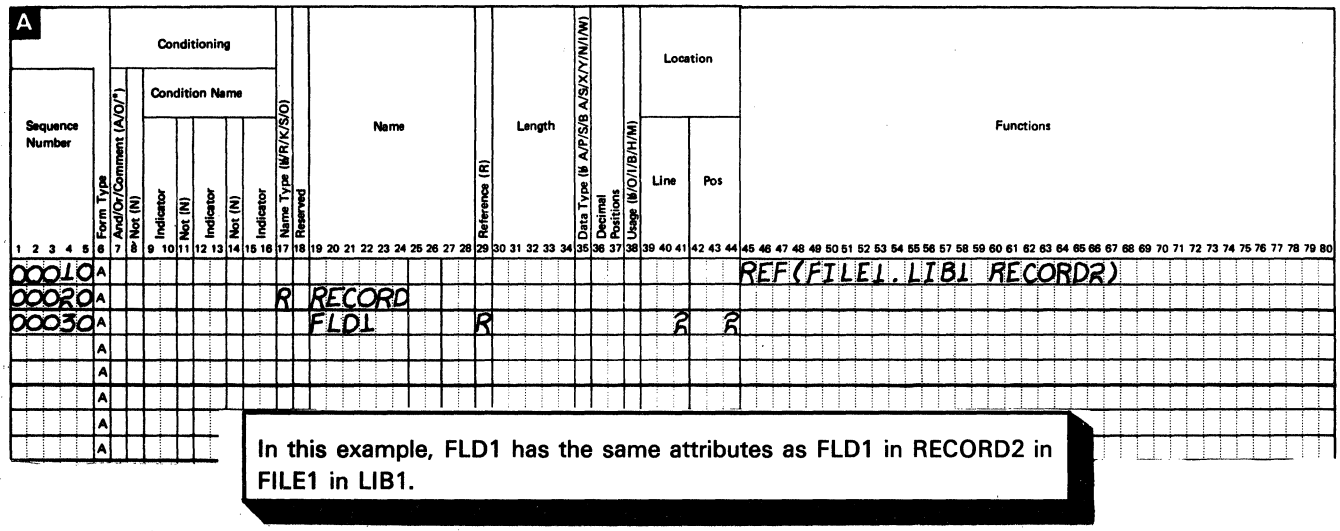
Option indicators are not valid with this keyword.

The following examples show how to specify the REF keyword:



Seq Number	Form Type	And/Or/Comment (A/O/*)	Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blR/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blA/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blO/I/B/H/M)	Location Line	Location Pos	Functions
00010	A																		REF(FILE1)
00020	A								R		RECORD								
00030	A										FLD1	R					2	2	
	A																		
	A																		
	A																		
	A																		
	A																		

In this example, FLD1 has the same attributes as the first (or only) FLD1 in FILE1.



Seq Number	Form Type	And/Or/Comment (A/O/*)	Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blR/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blA/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blO/I/B/H/M)	Location Line	Location Pos	Functions
00010	A																		REF(FILE1 LIB1 RECORD2)
00020	A								R		RECORD								
00030	A										FLD1	R					2	2	
	A																		
	A																		
	A																		
	A																		
	A																		

In this example, FLD1 has the same attributes as FLD1 in RECORD2 in FILE1 in LIB1.

REFFLD(referenced-field-name[.record-format-name] {*SRC / data-base-file-name [.library-name]})

Use this field level keyword when referencing a field under one of these three conditions:

- The name of the referenced field is different from the name in positions 19 through 28.
- The name of the referenced field is the same as the name in positions 19 through 28, but the record format, file, or library of the referenced field is different from that specified with the REF keyword.
- The referenced field occurs in the same DDS source file as the referencing field.

The referenced field name is required even if it is the same as the referencing field. Use the record format name when the referenced file contains more than one record format. Use *SRC to search the DDS source file in which the referencing field occurs (this is the default value when the data base file name and library name are not specified). Specify the data base file name (qualified by its library name, if necessary) when you want to search a particular data base file.

If, in the same DDS source file, you specify REF at the file level and REFFLD at the field level, the REFFLD specification is used. The particular search sequence depends on both the REF and REFFLD keywords. For more information, see Appendix A, *How to Specify REF and REFFLD.*

An R must be specified in position 29.

Option indicators are not valid with this keyword.

The following example shows how to specify the REFFLD keyword. Because the REF keyword is not specified, the default for lines 00030 and 00040 is to search the DDS source file in which they are specified. In line 00080, the parameter value *SRC explicitly specifies the source file. See the example in Appendix A, *How to Specify REF and REFFLD*, for explanations of the specifications.

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A								R		RECORD1								
00020	A										ITEM		5					1	
00030	A										ITEM1	R						2	REFFLD(ITEM)
00040	A										ITEM2	R						1 2	REFFLD(ITEM.FMAT1)
00050	A										ITEM3	R						2 2	REFFLD(ITEM FILEX)
00060	A										ITEM4	R						3 2	REFFLD(ITEM FILEX.LIBY)
00070	A										ITEM5	R						4 2	REFFLD(ITEM.FMAT1 FILEX.LIBY)
00080	A										ITEM6	R						5 2	REFFLD(ITEM *SRC)
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

SKIPA(skip-after-line-number)

Use this file, record, or field level keyword to specify that the printer device is to skip to a specific line number after it prints one or more lines. The parameter value is required and must be in the range 1 through 255.

If you specify the keyword at the file level, you must option it with one or more indicators; otherwise, option indicators are optional. The specified skip is performed after each record in the file has been printed.

If you specify the keyword at the record level, skipping is performed after all the lines associated with the record have been printed and before any file level SKIPA keywords have been applied.

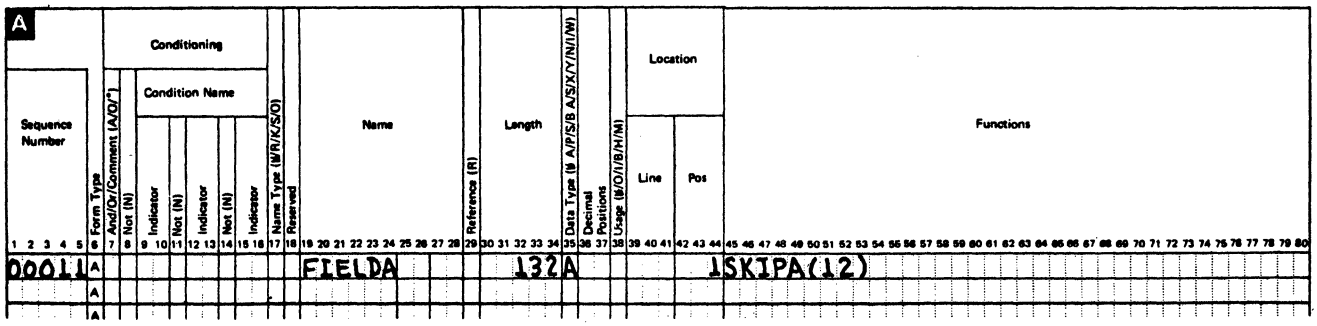
If you specify the keyword at the field level, skipping is performed after the field is printed.

Note: If line numbers are not used and you do not specify skip or space keywords, overprinting can result.

This keyword can be specified once at the file level, once at the record level, and once for each field.

This keyword is valid at the file level for all records, but it is not valid at the record level or the field level for records that have line numbers specified (positions 39 through 41). (The line number entries are flagged as errors.)

The following example shows how to specify the SKIPA keyword:



SKIPB(skip-before-line-number)

Use this file, record, or field level keyword to specify that the printer device is to skip to a specific line number before it prints the next line(s). The parameter value is required and must be in the range 1 through 255.

If you specify this keyword at the file level, you must option it with one or more indicators; otherwise, option indicators are optional. The specified skip is performed before each record in the file is printed and after applying any file level SKIPB operations.

If you specify this keyword at the record level, skipping is performed before any of the lines associated with that record are printed.

If you specify this keyword at the field level, skipping is performed before the field is printed.

This keyword can be specified once at the file level, once at the record level, and once for each field.

This keyword is valid at the file level for all records, but it is not valid at the record or field level for records that have line numbers specified (positions 39 through 41). (The line numbers are flagged as errors.)

Note: If line numbers are not used and you do not specify skip or space keywords, overprinting can result.

The following example shows how to specify the SKIPB keyword:

A

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00022	A								R		RFMTPR								SKIPB(5)
	A																		
	A																		

SPACEA(space-after-value)

Use this record or field level keyword to specify that the printer device is to space some number of lines after it prints one or more lines. The parameter value is required and must contain one of the following values:

0 – No spacing
1 – Space one line
2 – Space two lines
3 – Space three lines

If you specify this keyword at the record level, spacing occurs after all lines associated with that record have been printed. This keyword can be specified only once at the record level and once for each field.

If you specify this keyword at the field level, spacing is performed after the field is printed.

This keyword is not valid for records that have line numbers specified (positions 39 through 41). (The line numbers are flagged as errors.)

Option indicators are valid with this keyword.

Note: If line numbers are not used and you do not specify space or skip keywords, overprinting can result.

The following example shows how to specify the SPACEA keyword:

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (M/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
001000	A								R		RFMTPR								SPACEA(1)
001001	A										FIELDA		132					1	
	A																		
	A																		

SPACEB(space-before-value)

Use this record or field level keyword to specify that the printer device is to space some number of lines before printing the next line or lines. The parameter value is required and must contain one of the following values:

0 – No spacing
1 – Space one line
2 – Space two lines
3 – Space three lines

If you specify the keyword at the record level, spacing is performed before any lines associated with that record are printed. If you specify it at the field level, spacing is performed before the line containing that field is printed. This keyword can be specified only once at the record level or once for each field.

This keyword is not valid for records that have line numbers specified (positions 39 through 41). (The line numbers are flagged as errors.) Option indicators are valid with this keyword.

Note: If line numbers are not used and you do not specify space or skip keywords, overprinting can result.

The following example shows how to specify the SPACEB keyword:

A

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Conditioning: Not (N) (8)	Indicator (9–10)	Not (N) (11)	Indicator (12–13)	Not (N) (14)	Indicator (15–16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19–28)	Reference (R) (29)	Length (30–34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36–37)	Usage (b/O/I/B/H/M) (38)	Location Line (39–41)	Location Pos (42–44)	Functions (45–80)
00010	A										FIELDA		25	A				55	SPACEB(3)
00011	A										FIELDB		30					100	
	A																		
	A																		

TEXT('description')

Use this record or field level keyword to supply a text description (or comment) for the record format or field. This text is used for program documentation. If the length of the text is greater than 50 positions, only the first 50 characters are used by the high-level language compiler.

The TEXT keyword is valid for any record format or named field.

Option indicators are not valid with this keyword.

The following example shows how to specify the TEXT keyword:

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (M A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (M/O/I/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
0001.00	A								R		CUSMST								TEXT('Customer Master Record')
0002.00	A										FLD1		3		0		2		TEXT('Order Number')
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

TIME

Use this field level keyword to specify that CPF is to retrieve the current system time and print it in a constant field. The time is retrieved when the printer file is opened and remains the same until the file is closed. The following information is implied:

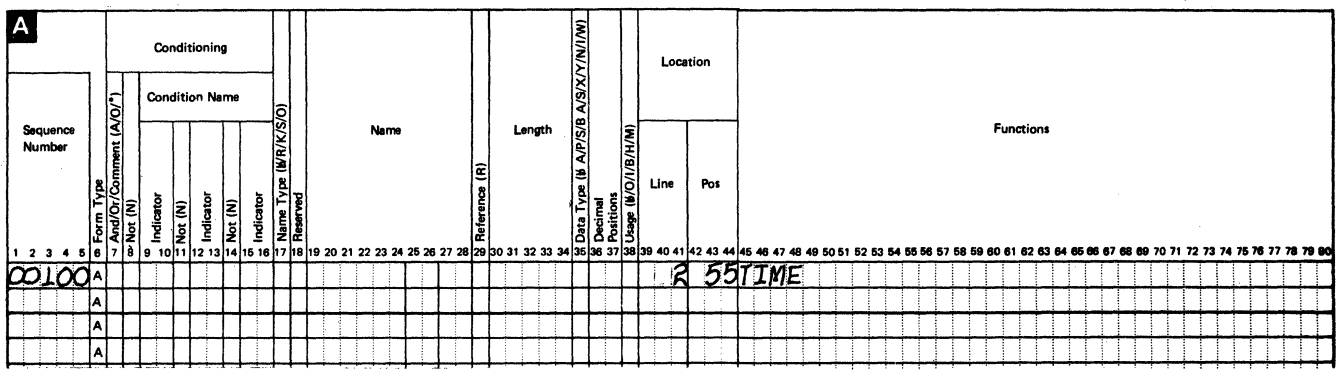
- A print length of 8 positions
- The edit word '0b:b b:b b' (if no editing specified, this edit word is assumed)

You can specify only the location of the field on the page. Positions 17 through 38 must be blank.

You can also specify a different edit word; for example, EDTWRD('0bHRS:b bMINS:b bSECS.') or one of the user-defined edit codes to be used in place of the implied edit word (see EDTCDE and EDTWRD keywords).

Option indicators are not valid with this keyword.

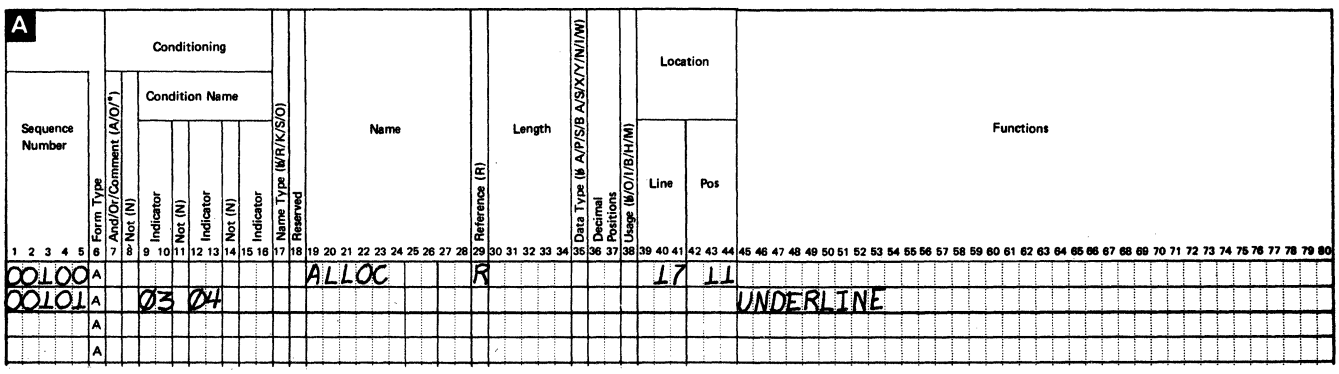
The following example shows how to specify the TIME keyword:



UNDERLINE

Use this field level keyword to specify that CPF is to underline the field when it is printed. UNDERLINE can be specified only if the printer supports underlining. Option indicators are valid with this keyword.

The following example shows how to specify the UNDERLINE keyword:



Chapter 6. Communications Files and BSC Files

This chapter is divided into two sections. The first, *Positional Entries*, provides rules and examples for filling in positions 1 through 44 of the Data Description Specifications form. To find information in this section, first determine what position on the form to use, then look up the section describing that position.

The second section of this chapter, *Keyword Entries*, provides rules and examples for specifying DDS keywords. The keywords are described in alphabetical order.

Complete communications and BSC (Binary Synchronous Communications) file examples can be found in Appendix F, *Examples*.

An explanation of *file level, record level,* and *field level* can be found in Chapter 1, *Introduction.*

The syntax rules for specifying DDS keywords can be found in Chapter 1, *Introduction.*

For guidance in choosing positional entries and keywords for communications or BSC files, see the *Data Communications Programmer's Guide.*

When you are describing a communications or a BSC file, you must specify the entries in the following order:

1. File level entries (optional)
2. Record level entries
3. Field level entries

} Repeat these entries for each record format in the file.

You must specify at least one record format in the file and at least one field in the file.

Note: The file name is specified through the Create Communications File (CRTCMNF) command or the Create BSC File (CRTBSCF) command, not through DDS.

The maximum number of record formats in a communications or a BSC file is 1024. The maximum number of fields in any one record format is 32 767 for communications files and 8192 for BSC files. For communication files, the maximum combined length of all *fields and indicators* in the buffer for one record format is 32 767. For BSC files, the maximum combined length of all *fields* in the buffer for one record format is 8192.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050*
Printed in U.S.A.

File		Keying Instruction	Graphic								Description	Page	of
Programmer	Date		Key										

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A	*		COMMUNICA		TIONS FILE					OR BSC				FILE				CODING EXAMPLE
00020	A	*																	
00030	A								R		RCD1								RCVENDGRP(14)
00040	A										FLDA		5						
00050	A										FLDB		5		Ø				
00060	A										FLDC		10		2				
00070	A																		
00080	A								R		RCD2								
00090	A			72		73													
00110	A	ON		74															FAIL
00120	A										FLDD		12						
	A										FLDC	R							
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

*Number of sheets per pad may vary slightly.

Figure 6-1. Communications or BSC File Coding Example

Positional Entries (Positions 1-44)

This section describes how to specify the first 44 positions of the Data Description Specifications form for communications and BSC files. To code the rest of the form, see the section *Keyword Entries (Functions Field, Positions 45-80)* later in this chapter.

Figure 6-1 shows some of the positional entries for communications files and BSC files.

Positions 1-5 (Sequence Number)

Use these positions to specify a sequence number for each line on the form. The sequence number is optional and is used for documentation purposes only.

Position 6 (Form Type)

The A in this position designates this as a Data Description Specifications form. The form type is optional and is for documentation purposes only.

Position 7 (Comment)

An asterisk (*) in position 7 identifies the line as a comment line. Comment lines can appear anywhere in the DDS and are retained only in the source file and printed on the source listing. Use positions 8 through 80 for comment text. A blank line (no characters specified in positions 7 through 80) is treated as a comment line.

Positions 7-16 (Conditioning)

Positions 7 through 16 are a multiple-field area in which you can specify option indicators. Option indicators are 2-digit numbers from 01 to 99. Your program can set option indicators on or off to select a keyword for output operations. *Option indicators are not valid for fields in communications and BSC files.* (Response indicators can be specified only as parameter values for the keywords INDTXT, RCVENDGRP, and TRNRND. Figure 6-1 shows how to specify option and response indicators.)

A condition is an ANDed grouping of one to nine indicators that must all be in effect (set off if N is not specified; set on if N is specified) before the keyword is selected.

A maximum of nine option indicators for each condition and nine conditions for each keyword is valid; therefore, a maximum of 81 indicators can be specified for each keyword, when nine indicators are used with nine conditions.

When you specify a condition so that more than one indicator must be on or off before the condition is satisfied, it is called an AND condition; you are essentially saying that the first indicator you specified, AND the second, AND the third, and so on, must be all in effect before the condition is satisfied and the keyword is selected. The keyword must be specifed on the same line as the last (or only) set of indicators specified.

You can also specify several conditions for a keyword so that if any one of them is satisfied, the keyword is selected. This is called an OR relationship, and it means that if the first condition is satisfied, OR the second condition, OR the third condition, and so on, the keyword is to be selected. Note that conditions within the OR relationship can consist of just one indicator or can consist of several indicators ANDed together. Indicators can be ANDed to form a condition; conditions can be ORed to give your program several ways to select the keyword.

AND (Position 7): If you need more than three indicators to form an ANDed condition, specify them on the next line or lines. You can specify A in position 7 on the second or following lines to designate the continuation of the ANDed condition; or you can leave it blank because A is the default.

OR (Position 7): If you specify several conditions that are to be ORed together, each condition must start on a new line and each condition except the first must have an O in position 7. An O specified for the first condition produces a warning message, and that position is assumed to be blank.

NOT (Positions 8, 11, 14): If you want an indicator to be off instead of on to satisfy a condition, specify N in the position just preceding the indicator (position 8, 11, or 14).

Conditioning More Than One Keyword: If you want to condition one or more keywords, specify the keywords on the same line. The last (or only) indicator must appear on the same line as the keywords. If the conditioning applies to keywords on more than one line, keyword continuation must be used for the indicators to apply to all keywords. See the *Data Description Specifications Syntax Rules* in Chapter 1, *Introduction.*

Position 17 (Name Type)

The value in this position identifies the type of name specified in positions 19 through 28 (name). The valid entries for communications files and BSC files are:

Entry	Meaning
R	Record format name
Blank	Field name

Figure 6-1 shows how to code the name type field.

Position 18 (Reserved)

This position does not apply to any file and must be blank unless an asterisk appears in position 7 (which indicates a comment line).

Positions 19-28 (Name)

Use these positions to specify record format names and field names. The following rules apply:

- Names must be 10 characters or less.
- Names must start in position 19.
- Names must begin with an alphabetic character (A through Z, @, $, and #). All subsequent characters can be alphameric (A through Z, 0 through 9, @, $, #, and underscore [__]). There can be no embedded blanks.

High-level languages can impose specific length and value restrictions on names. For example, RPG III accepts only field names of 6 characters or less and record format names of 8 characters or less, and they cannot contain an underscore. It is your responsibility to make sure that the names used here are acceptable to all language processors that process the file.

Record Format Name: When you specify R in position 17 (name type), the name specified in positions 19 through 28 is a record format name. You can specify more than one record format for a communications file or BSC file, but each record format name must be unique within that file. You must specify field names to complete a record format in a communications file or a BSC file.

Field Name: When you specify a blank in position 17 (name type), the name specified in positions 19 through 28 is a field name. Field names must be unique within the record format. For communications and BSC files, the order in which field names are specified in the DDS is the order the fields take in the input and output buffers (the input and output buffers are the same).

Position 29 (Reference)

Specify R in this position to use the reference function of CPF to copy the attributes of a previously defined named field (called the *referenced field*) in the field you are defining now. The referenced field can be previously defined in the communications or the BSC file you are defining; or it can be in a previously created data base file (the data base file to be referenced is specified on the REF or REFFLD keywords). The field attributes referenced are the length, the data type, and the decimal positions of the field, as well as editing and text keywords.

If you do not specify R, you cannot use the reference function for this field and you must specify field attributes for this field.

Position 29 must be blank at the file and record levels.

The name of the referenced field can be either the same as the field you are defining or different from the field you are defining. If it is the same, you need only specify R in position 29 (in addition to specifying the name of the field you are defining in positions 19 through 28). If it is different, you must specify the name of the referenced field with the REFFLD keyword.

You can specify the name of the file defining the referenced field as a parameter value with the REF or the REFFLD keyword. See REF and REFFLD keyword descriptions later in this chapter and Appendix A, *How to Specify REF and REFFLD*, for explanations of how CPF finds the referenced field.

You do not need to copy all attributes from a previously defined field to the field you are defining. To override specific attributes of the referenced field, specify those attributes for the field you are defining. For example, if you specify length for the field you are defining, the length is not copied from the referenced field.

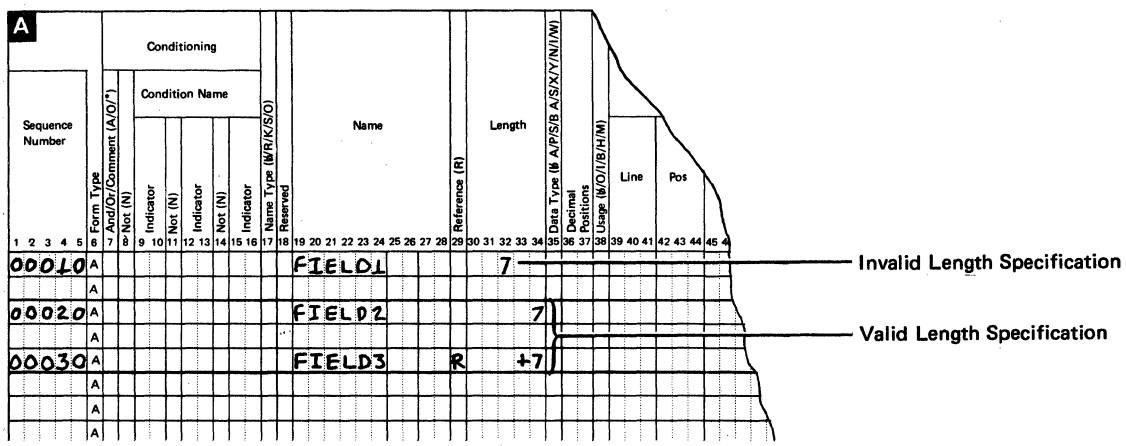
When you override data type to character (by specifying A in position 35), decimal positions are not copied from the referenced field.

Note: Once the communications or the BSC file is created, the referenced file can be deleted or changed without affecting the field definitions in the communications or the BSC file. To incorporate changes made in the referenced file, delete and re-create the communications or the BSC file.

Positions 30-34 (Length)

You must specify a length for each named field unless the length is being duplicated from a referenced field. Specify the number of digits for a numeric field, or the number of characters for a character field. The length specification must be right-justified; leading zeros are optional.

The following example shows valid and invalid length specifications:



Valid length specifications are:

	Valid Length	
Data Type	**Communications Files**	**BSC Files**
Character	1 through 32 767 characters	1 through 8192 characters
Binary	1 through 9 digits	1 through 9 digits
Zoned decimal	1 through 31 digits	1 through 31 digits
Packed decimal	1 through 31 digits	1 through 31 digits

The sum of the number of bytes occupied by all fields in a record must not exceed 32 767 for communications files and 8192 for BSC files. The number of bytes actually occupied is determined by the system as follows:

Data Type	Bytes Occupied
Character	Number of characters
Binary	
1-4 digits	2 bytes
5-9 digits	4 bytes
Zoned decimal	Number of digits
Packed decimal	(Number of digits/2) + 1

Figure 6-1 shows how to code the field length.

If you are using a referenced field, you can override the length of the field by specifying a new value or by specifying the increase or decrease in length. To increase the length, specify +n, where n is the increase. To decrease the length, specify -n, where n is the decrease. For example, an entry of +4 for a numeric field indicates that is to be 4 digits longer than the referenced field.

Note: High-level languages can impose specific length and value restrictions on the field length; these restrictions should be observed for files used by those languages.

Position 35 (Data Type)

Use this position to specify the data type of the field within the file. The valid data type entries for communications and BSC files are:

Entry	Meaning
Numeric types	
P	Packed decimal
S	Zoned decimal
B	Binary
Alphabetic type	
A	Character

If you do not specify a data type and do not duplicate one from a referenced field, CPF assigns a default value as follows:

- A (character) if the decimal positions (36 through 37) are blank
- S (zoned decimal) if the decimal positions (36 through 37) contain a number in the range 0 through 31

Note: Placing 0 in position 37 is a convenient way to specify an integer numeric field.

Figure 6-1 shows how to code the data type.

Positions 36-37 (Decimal Positions)

Use these positions to specify the decimal placement within a packed decimal, a zoned decimal, or a binary field. Specify a decimal number from 0 through 31 to indicate the number of decimal positions to the right of the decimal point. (This number must not be greater than the number of digits specified in the field length.) Figure 6-1 shows how to code the decimal positions field.

If you are using a referenced field, these positions can be either overridden or changed. To override the positions, specify the new value. To change the positions, specify the amount you want the field increased or decreased and precede it with either a + or -, respectively. For example, an entry of +4 indicates there are to be four more digits to the right of the decimal point than were in the referenced field. If the resulting number of decimal positions is greater than the maximum allowed, an error message is sent.

Note: High-level languages can impose specific length and value restrictions on the decimal positions; these restrictions should be observed for files used by those languages.

Position 38 (Usage)

For communications files or BSC files, you can specify B (for both, or output/input fields); if left blank, the default is B.

Positions 39-44 (Location)

These positions do not apply for a communications or a BSC file and must be blank unless an asterisk appears in position 7.

Keyword Entries, (Positions 45-80)

This section contains keyword entries valid for defining communications files and BSC files. They are entered in positions 45 through 80 (functions). See *Data Description Specifications Syntax Rules* in Chapter 1 for a discussion of the general rules for specifying keywords. Figure 6-2 shows how to use the general rules to specify DDS for a communications file or BSC file.

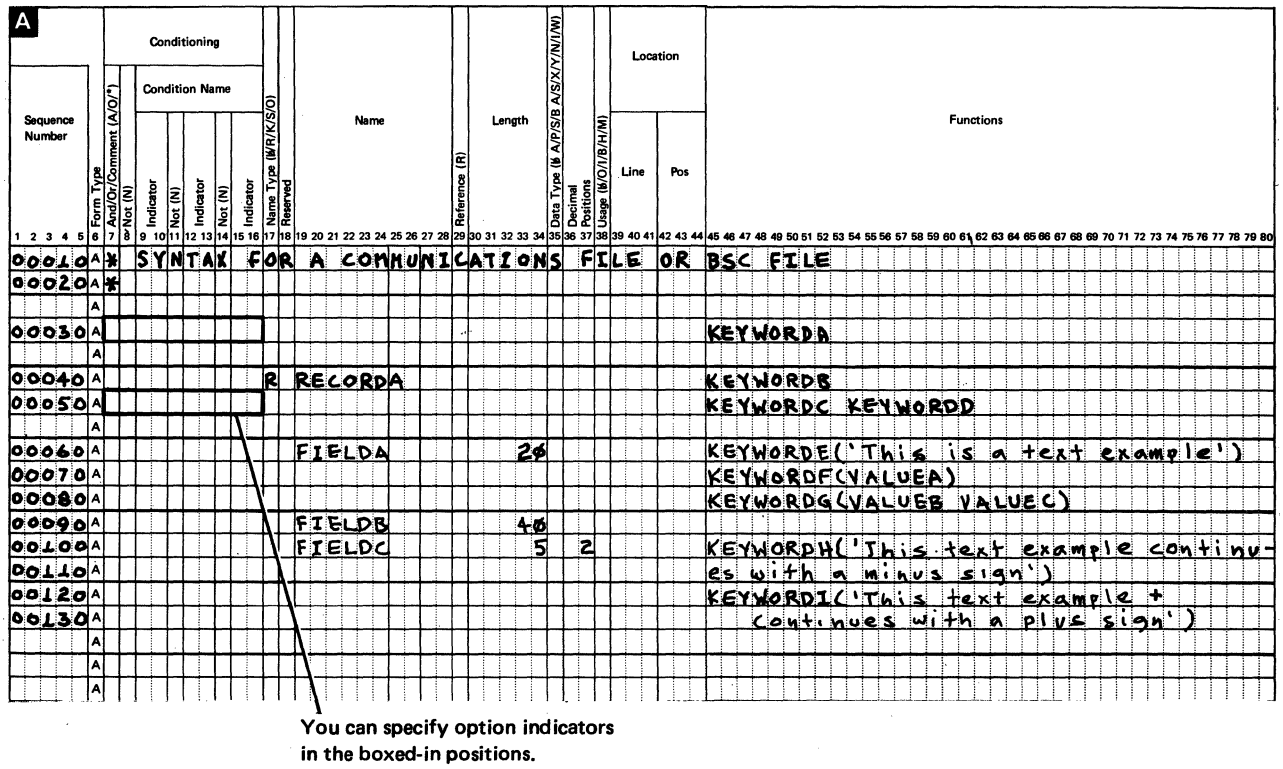


Figure 6-2. Syntax for a Communications File or BSC File

In this section, keywords are described in alphabetical order, and each is illustrated with an example.

The following keywords are valid for communications files and BSC files:

ALWWRT
ENDGRP
FAIL
INDTXT
RCVENDGRP
REF
REFFLD
RQSWRT
RTGDEV
RTGDEVCLS
RTGFMT
RTGPOS
TEXT
TRNRND

ALWWRT

Use this file or record level keyword (allow write) so that your program can indicate that it has finished sending. The ALWWRT keyword is valid only on an output operation when your program is in the send state.

Option indicators are valid with this keyword. (When you specify this keyword at the file level, you should specify an option indicator.)

SNA Considerations: CPF sends an end chain (EC bit) and a change direction indication (CD bit). Your program goes to the receive state.

BSC Considerations: CPF sends an end of transmission (EOT). Your program goes to the contention state.

The following example shows how to specify the ALWWRT keyword:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: or	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9	10	11 12 13	14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
01000	A			21																ALWWRT
02000	A									R		CUSMST								
	A																			
	A																			
	A																			
	A																			

ENDGRP

Use this file or record level keyword (end of group) so that your program can indicate the end of a user-defined group of records. The ENDGRP keyword is valid only on an output operation when your program is in the send state. (Your program remains in the send state.)

Option indicators are valid with this keyword. (When you specify this keyword at the file level, you should specify an option indicator.)

SNA Considerations: CPF sends an end chain (EC bit).

BSC Considerations: CPF sends a null record (STXETX). When your program transmits user-blocked data (parameter BLOCK(*USER) specified on the Create BSC File (CRTBSCF), the Change BSC File (CHGBSCF), or the Override BSC File (OVRBSCF) command), the ENDGRP and RCVENDGRP keywords have no effect.

The following example shows how to specify the ENDGRP keyword:

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9–10)	Not (N) (11)	Indicator (12–13)	Not (N) (14)	Indicator (15–16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19–28)	Reference (R) (29)	Length (30–34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36–37)	Usage (b/O/I/B/H/M) (38)	Location Line (39–41)	Location Pos (42–44)	Functions (45–80)
00030	A								R		RECORD1								ENDGRP
	A																		
	A																		
	A																		

FAIL

Use this file or record level keyword so that your program can terminate a session when it can no longer continue because of some problem. All records not already sent in the current transmission are discarded, and no further records are sent. The FAIL keyword is valid only on an output operation; your program can issue an output operation with the FAIL keyword selected when it is in either the receive or the send state.

If the FAIL keyword is in effect, the keywords ALWWRT, ENDGRP, and RQSWRT are ignored.

Option indicators are valid with this keyword. (When you specify this keyword at the file level, you should specify an option indicator.)

SNA Considerations: In the receive state, when your program issues an output operation with the FAIL keyword in effect, an SNA negative response (-RSP) is transmitted.

In the send state, when your program issues an output operation with the FAIL keyword in effect, an SNA CANCEL is transmitted.

BSC Considerations: In the receive state, when your program issues an output operation with the FAIL keyword in effect, an abort, which is an end of transmission (EOT), is transmitted instead of the normal acknowledgment (ACK0 or ACK1).

In the send state, when your program issues an output operation with the FAIL keyword in effect, a forward abort sequence (STXENQ followed by an EOT) is transmitted.

The following example shows how to specify the FAIL keyword:

A

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00010	A								R		INQ								
00020	A			99															FAIL
	A																		
	A																		

INDTXT(response-or-option-indicator 'indicator-usage-text')

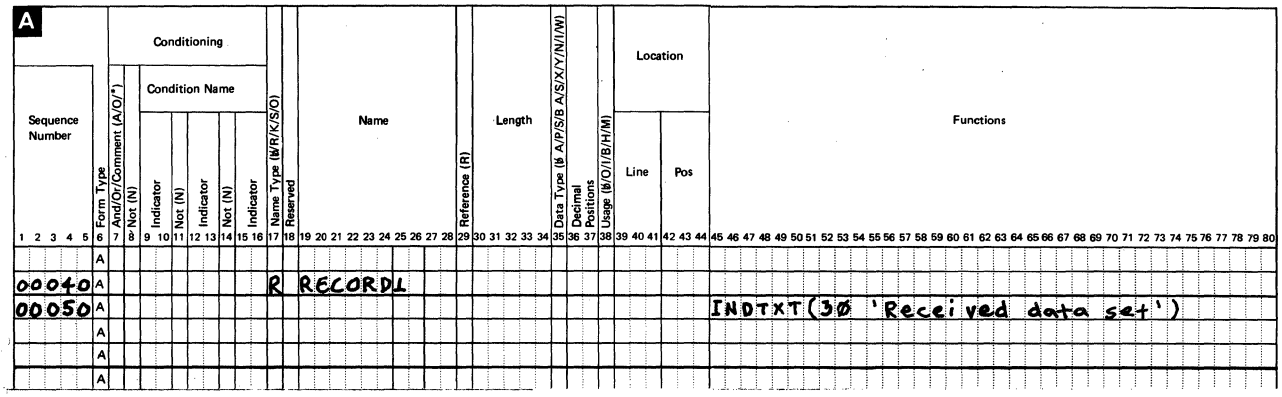
Use this file or record level keyword to associate descriptive text (indicating intent or usage) with a specific response or option indicator. You can specify it once for each response and option indicator.

If you specify this keyword, indicator usage text is a required parameter value and must be a character literal enclosed in apostrophes. If the length of the literal is greater than 50 positions, only the first 50 characters are used by the high-level language compiler. The text is used during compilation to aid program documentation.

Option indicators are not valid with this keyword.

Note: The INDTXT keyword does not cause the specified indicator to appear in either the input or the output record area. It provides text to be associated with the indicator. Once an indicator has been given a textual assignment (either by this keyword or by the response indicator text), no other textual assignment is made. A message is issued and the keyword is ignored. This differs from other keywords that can have indicators specified as parameter values, in that, for other keywords, the text is ignored.

The following example shows how to specify the INDTXT keyword:



RCVENDGRP(response-indicator ['text'])

Use this file or record level keyword (receive end of group) so that CPF can set on the specified response indicator to inform your program of the end of a user-defined group of records. The RCVENDGRP keyword is valid only on an input operation when your program is in the receive state. (Your program remains in the receive state.)

The optional text is included on the listing generated at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The apostrophes are required. If you specify more than 50 characters between the apostrophes, the text is truncated to 50 characters on the program listing.

Option indicators are not valid for this keyword.

SNA Considerations: The response indicator is set on when CPF detects an end chain (EC bit).

BSC Considerations: The response indicator is set on when CPF detects a null record (STXETX). When receiving user-blocked data (parameter BLOCK(*USER) specified on the Create BSC File (CRTBSCF), the Change BSC File (CHGBSCF), or the Override BSC File (OVRBSCF) command), the RCVENDGRP keyword has no effect.

The following example shows how to specify the RCVENDGRP keyword:

A

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9–10)	Not (N) (11)	Indicator (12–13)	Not (N) (14)	Indicator (15–16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19–28)	Reference (R) (29)	Length (30–34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36–37)	Usage (b/O/I/B/H/M) (38)	Location Line (39–41)	Location Pos (42–44)	Functions (45–80)
00100	A								R		CUSMST								
00200	A																		RCVENDGRP(66 'End of group received-
	A																		')
	A																		
	A																		
	A																		

REF(data-base-file-name[.library-name][record-format-name])

Use this file level keyword to specify the name of a file from which field descriptions are to be retrieved. Use it when you want to duplicate descriptive information from several fields in a previously described record format. You can code the file name once rather than on REFFLD keywords with each of the field descriptions that reference the file. To reference more than one file, use the REFFLD keyword. (The REF keyword can be specified only once.)

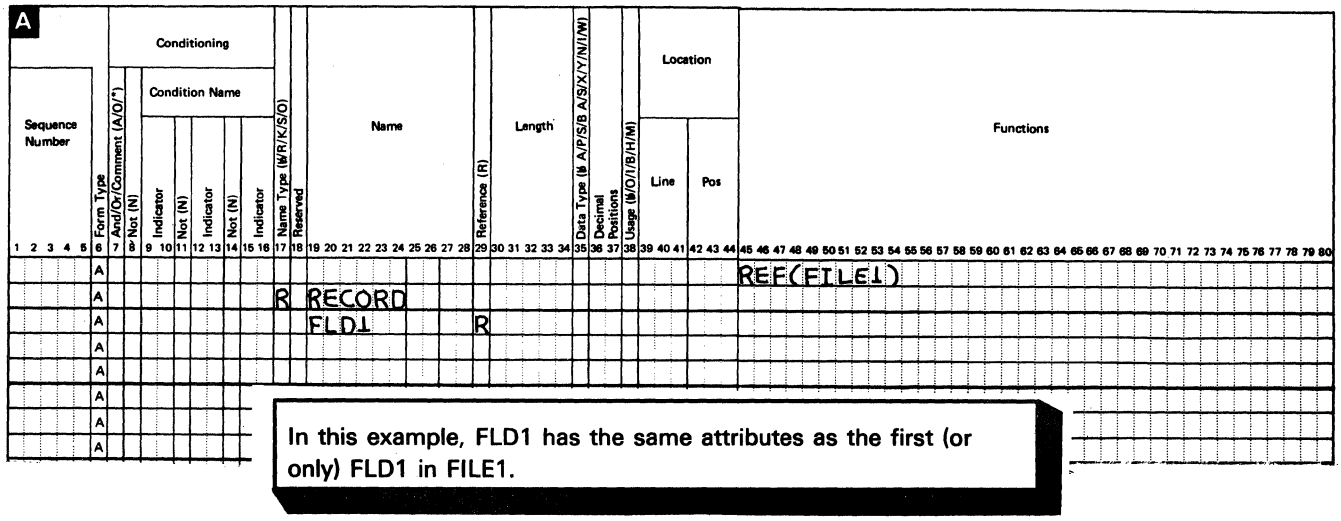
If there is more than one record format in the referenced file, specify a record format name as a parameter value for this keyword to tell CPF which one to use unless the record formats should be searched sequentially.

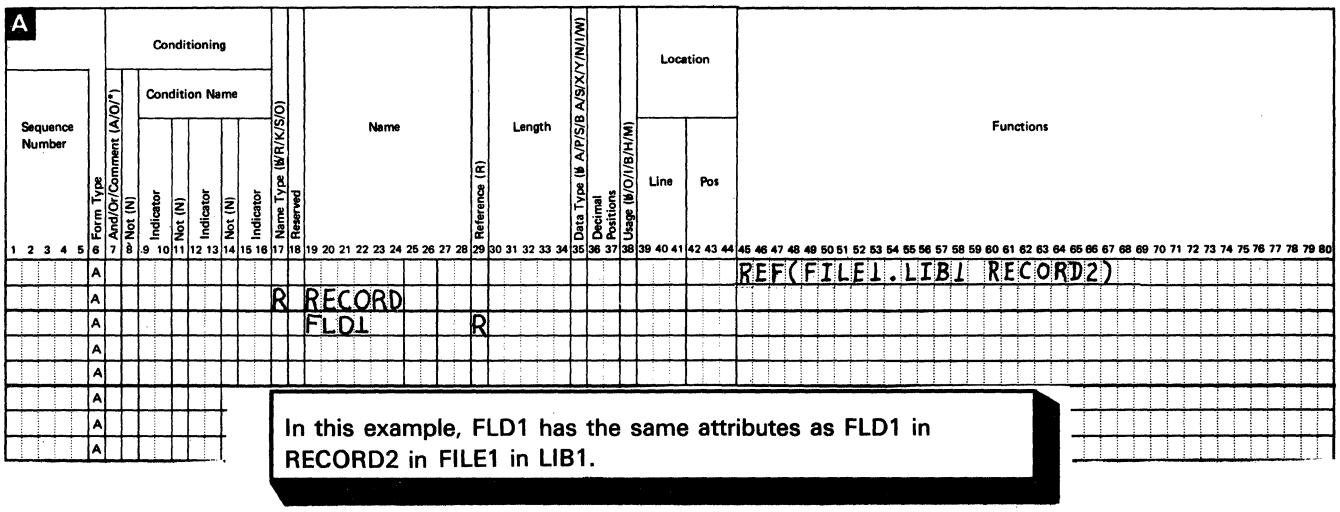
The file name is a required parameter value for this keyword; the library name and the record format name are optional.

If you do not specify the library name, the current library list is used. If the record format name is not specified, each format is searched in order (as they are specified); the first occurrence of the field name is used. For more information, see Appendix A, *How to Specify REF and REFFLD.*

Option indicators are not valid with this keyword.

The following examples show how to specify the REF keyword:





REFFLD(referenced-field-name[.record-format-name] {*SRC / [data-base-file-name [.library-name]]})

Use this field level keyword when referencing a field under one of these three conditions:

- The name of the referenced field is different from the name in positions 19 through 28.
- The name of the referenced field is the same as the name in positions 19 through 28, but the record format, file, or library of the referenced field is different from that specified with the REF keyword.
- The referenced field occurs in the same DDS source file as the referencing field.

The referenced field name is required even if it is the same as the referencing field. Use the record format name when the referenced file contains more than one record format. Use *SRC to search the DDS source file in which the referencing field occurs (this is the default value when the data base file name and the library name are not specified). Specify the data base file name (qualified by its library name if necessary) when you want to search a particular data base file.

If, in the same DDS source file, you specify REF at the file level and REFFLD at the field level, the REFFLD specification is used. The particular search sequence depends on both the REF and REFLD keywords. For more information, see Appendix A, *How to Specify REF and REFFLD.*

An R must be specified in position 29.

Option indicators are not valid with this keyword.

The following example shows how to specify the REFFLD keyword. Because the REF keyword is not specified, the default for lines 00030 and 00040 is to search the DDS source file in which they are specified. In line 00080, the parameter value *SRC explicitly specifies the source file. See the example in Appendix A, *How to Specify REF and REFFLD*, for explanations of the specifications.

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: or	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/I/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16			17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A									R		RECORD1								
00020	A											ITEM		5						
00030	A											ITEM1	R							REFFLD(ITEM)
00040	A											ITEM2	R							REFFLD(ITEM.FMAT1)
00050	A											ITEM3	R							REFFLD(ITEM FILEX)
00060	A											ITEM4	R							REFFLD(ITEM FILEX.LIBY)
00070	A											ITEM5	R							REFFLD(ITEM.FMAT1 FILEX.LIBY)
00080	A											ITEM6	R							REFFLD(ITEM *SRC)
	A																			
	A																			
	A																			

RQSWRT

Use this file or record level keyword (request write) so that your program can request permission to send data. The RQSWRT keyword is valid only on an output operation when your program is in the receive state.

If the RQSWRT keyword is in effect and the FAIL keyword is not in effect, the ALWWRT and ENDGRP keywords are ignored.

Option indicators are valid with this keyword. (When you specify this keyword at the file level, you should specify an option indicator.)

SNA Considerations: CPF sends a SIGNAL command.

BSC Considerations: CPF sends a reverse interrupt (RVI).

The following example shows how to specify the RQSWRT keyword:

A

Sequence Number (1-5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Not (N) (8)	Indicator (9-10)	Not (N) (11)	Indicator (12-13)	Not (N) (14)	Indicator (15-16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19-28)	Reference (R) (29)	Length (30-34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36-37)	Usage (b/O/I/B/H/M) (38)	Location Line (39-41)	Location Pos (42-44)	Functions (45-80)
00010	A								R		CUSMST								
00020	A			14															RQSWRT
	A																		
	A																		

Use of Routing Keywords

You can use the routing keywords (RTG prefix) to cause CPF to place specific information regarding input operations (up to 80 characters) in the routing identification (RTG) field in the I/O feedback area. No information is entered for an output-only operation.

CPF places the information into the routing field only after other specified functions have been performed successfully.

Option indicators are not valid for routing keywords.

Each routing keyword requires that you specify a *routing field location*, which is the first character position for that item in the routing field within the I/O feedback area. All entries must fit completely within the 80-byte routing area, and no checking is done to prevent overlapping entries in the routing area. If entries do overlap, the results can be predicted, based on the processing order of the keywords. Entries that are processed later overlay entries that are processed earlier.

Note: Multiple specifications are not checked for overlap in the I/O feedback area.

The processing order follows:

RTGDEVCLS
RTGDEV
RTGFMT
RTGPOS

Figure 6-3 shows how to specify the routing keywords for communications files and BSC files.

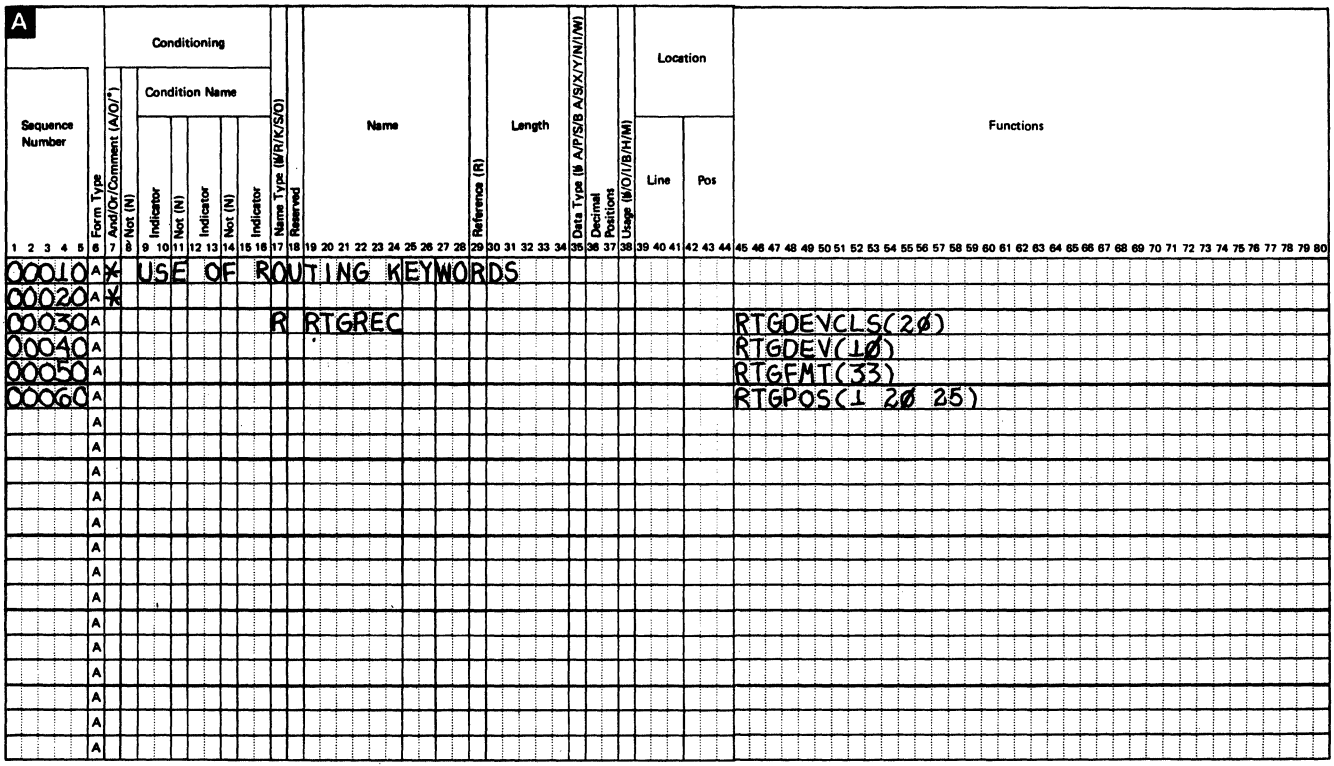


Figure 6-3. Routing Keywords Coding Example

RTGDEV(starting-routing-field-location)

This record level keyword causes CPF to place the 10-character device name into the routing field. Valid locations you can specify are within the range of 1 through 71.

RTGDEVCLS(starting-routing-field-location)

Use this record level keyword to cause CPF to place the 2-character device class indicator into the routing field. Valid locations you can specify are within the range of 1 through 79.

The class value that is placed in the routing field is hex 0609 for communications files and hex 070A for BSC files.

RTGFMT(starting-routing-field-location)

Use this record level keyword to cause CPF to place the record format name (as determined by CPF for the input operation) into the routing field. The length of this entry in the routing field is always 10 bytes.

Valid field locations you can specify are in the range of 1 through 71.

RTGPOS(starting-routing-field-location starting-data-position ending-data-position)

Use this record level keyword to identify character positions (within the data received from the host system) that are to be placed into the routing field. The positions are specified by supplying the starting position (relative to the beginning of the data stream) and ending position.

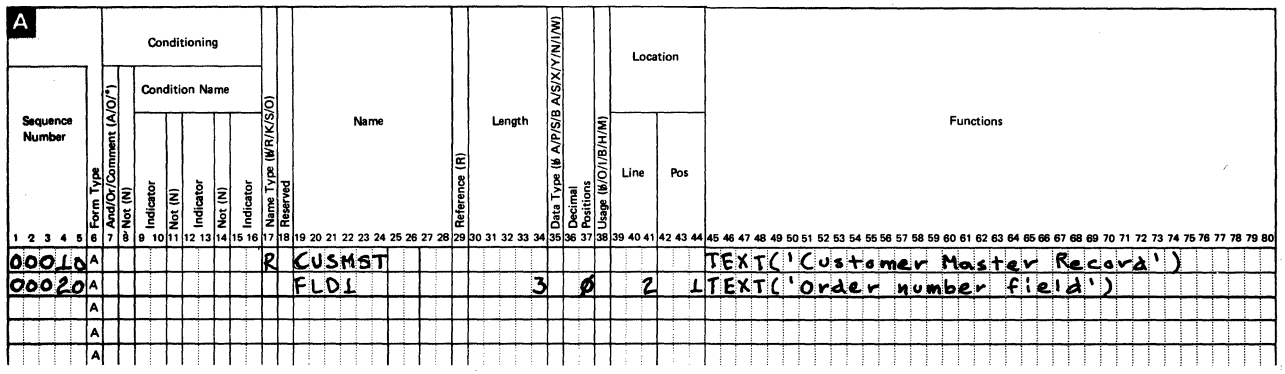
Valid field locations you can specify are within the range 1 through 80, depending on the starting and ending data locations.

TEXT('description')

The text information is useful in high-level language user program record and field documentation produced by compiler output. Use this record or field level keyword to supply a text description of the current field or record format. If the length of the text is greater than 50 positions, only the first 50 characters are used by the high-level language compiler. The text must be enclosed in apostrophes.

Option indicators are not valid with this keyword.

The following example shows how to specify the TEXT keyword:



A	Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (W/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (b A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (b/O/I/B/H/M)	Location: Line	Pos	Functions
	1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	00010	A								R		CUSMST								TEXT('Customer Master Record')
	00020	A										FLD1		3		Ø		2		TEXT('Order number field')
		A																		
		A																		
		A																		

TRNRND(response-indicator ['text'])

Use this file or record level keyword (turn around transmission) so that CPF can set on the specified response indicator to inform your program that the sending program or device has stopped sending. The TRNRND keyword is valid only on an input operation when your program is in the receive state.

The optional text is included on the listing generated at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The apostrophes are required. If you specify more than 50 characters between the apostrophes, the text is truncated to 50 characters on the program listing.

Option indicators are not valid with this keyword.

SNA Considerations: The response indicator is set on when CPF detects an end bracket (EB bit) or change direction (CD bit).

BSC Considerations: The response indicator is set on when CPF detects a normal end of transmission (EOT).

The following example shows how to specify the TRNRND keyword:

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Conditioning: Condition Name: Not (N)	Conditioning: Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/N)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A																		TRNRND(11 'Host has stopped sending-
00020	A																		')
00030	A								R		CUSMST								
	A																		
	A																		
	A																		

Appendix A. How to Specify REF and REFFLD

This appendix should help you decide whether to specify REF or REFFLD or both, and how to specify parameter values for each REF or REFFLD keyword you specify. Note that you must specify R in position 29 for each referencing field.

1. REF or REFFLD or both?

   If all or most of the referenced fields are defined in a single file other than the one you are now defining, specify REF at the file level. This should provide referencing for all or most of the referencing fields. (Note that the file you reference must be a data base file.)

   Specify REFFLD for every referencing field:

   - Not in the file specified on the REF keyword

   or

   - Whose name differs from the name of the field it references (this includes fields that reference fields in the file you are defining)

2. For each REFFLD keyword specified, do you need a data base file name?

   If the REF keyword is specified, any data base file name specified on the REFFLD keyword overrides the data base file name on the REF keyword. On the REFFLD keyword, you can specify:

   - *SRC so that CPF searches the file you are defining for the referenced field

   - The data base file name that CPF is to search for the referenced field

   If the REF keyword is not specified, you can specify:

   - *SRC so that CPF searches the file you are defining for the referenced field (this is the default)

   - The data base file name for CPF to search for the field

3. For each data base file name specified, do you need a library name?

   If the job that creates the file you are defining (perhaps, your interactive job) has a library list, and the specified data base file is on the library list, specify only the file name (FILE1). Otherwise, specify the file name qualified by the library name (FILE1.LIB1).

4. For each REF or REFFLD keyword specified, do you need a record format name?

   If the referenced file has only one record format, do not specify a record format name.

   If the referenced file has more than one record format, specify a record format name.

5. Repeat steps 2 through 4 for each REF or REFFLD keyword you specify.

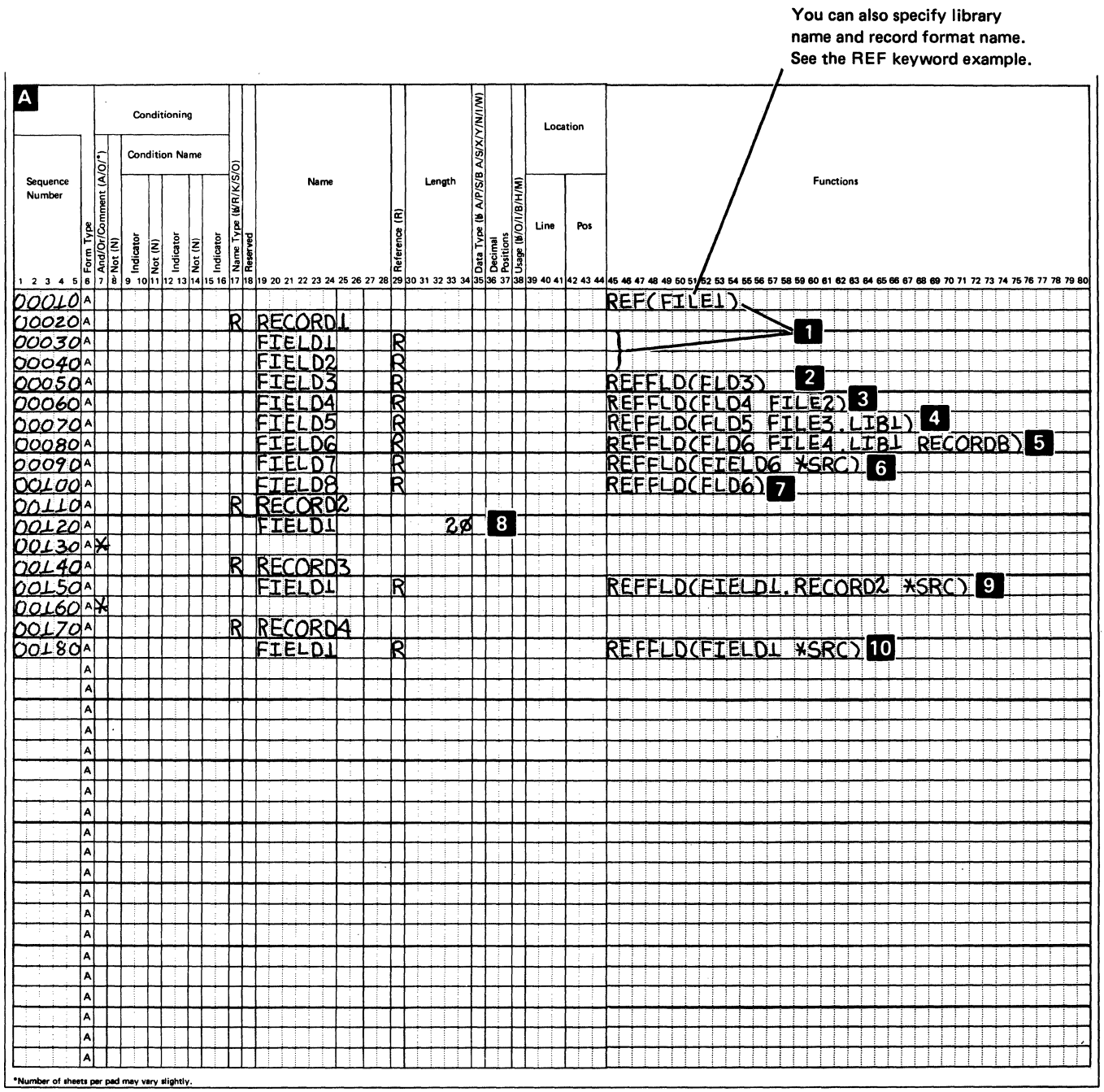


Figure A-1. Sample Reference Function Specifications

The following numbered items refer to Figure A-1:

1. FIELD1 and FIELD2 have the same attributes as FIELD1 and FIELD2 in FILE1.

2. FIELD3 has the same attributes as FLD3 in FILE1.

3. FIELD4 has the same attributes as FLD4 in FILE2 (different field name and different file from 1).

4. FIELD5 has the same attributes as FLD5 in FILE3 in LIB1.

5. FIELD6 has the same attributes as FLD6 in record format RECORDB in FILE4 in LIB1.

6. FIELD7 has the same attributes as FIELD6 (on the preceding line in this file).

7. FIELD8 has the same attributes as FLD6 in FILE1.

8. FIELD1 in RECORD2 has unique field attributes (it does not use the reference function; R is not specified in position 29).

9. FIELD1 in RECORD3 has the same attributes as FIELD1 in RECORD2.

10. FIELD1 in RECORD4 has the same attributes as FIELD1 in RECORD1.

Appendix B. Physical and Logical File Keyword Summary

Keyword	Values	Function	Level[1]	Groups[2]
ABSVAL		The sign of the field is ignored when sequencing values (use absolute values).	L/P-KFLD	PTH
ACCPTH	(data-base-file-name [.library-name])	The keyed access path of a previously created logical or physical file is shared.	L-FL	PTH
ALL		The system selects or omits all records not meeting the select/omit rules.	L-S/O FLD	S/O
ALTSEQ	(alternate-sequence-table-name [.library-name])	An alternate collating sequence is used for the key.	L/P-FL	PTH
CHECK	(AB, ME, MF, M10, M11, VN)	A field value must meet one or more of the following check algorithms to be valid: • Allow blank (AB) • Mandatory enter (ME) or fill (MF) • Valid name (VN) • IBM Modulus 10 (M10) or 11 (M11) self-check CHECK does not request any action by data base data management, but can be specified in the data base file so that it can be duplicated into the display file when the field is referenced at display file creation.	L/P-FLD	CHK

[1]Level:
L–Logical file
P–Physical file
FL–File level
RCD–Record level
FLD–Field level
KFLD–Key field level
S/O FLD–Select/omit field level

[2]Groups:
CHK–Validity checking keywords
EDT–Editing and field setting keywords
PTH–Access path keywords
REF–Naming and referencing keywords
S/O–Select/omit keywords
TXT–Text definition keywords

PHYSICAL AND LOGICAL FILE KEYWORDS (continued)

Keyword	Values	Function	Level[1]	Groups[2]
CMP		See COMP (the preferred spelling).		
COLHDG	('line-1' [line-2' ['line-3']])	This specifies a column heading for a field (used by query, DFU, and SDA).	L/P-FLD	TXT
COMP	(EQ, GE, GT, LE, LT, NE, NG, NL)	A field value must meet a comparision or relation such as *equal to*, to be valid or selected. COMP is also used as a validity checking keyword, specified for retention in the data base file so that it can be duplicated into a display file through the reference function.	L/P-FLD L-S/O FLD	CHK S/O
CONCAT	(field-1 field-2 [maximum])	Fields from a physical file are concatenated into a field in a logical file.	L-FLD	REF
DESCEND		Values in a key field are retrieved in descending sequence.	L/P-KFLD	PTH
DIGIT		Only the digit portion (low-order 4 bits) of each byte of the key field is used to build a key value.	L/P-KFLD	PTH
EDTCDE	(code [*|floating-currency-symbol])	This names the edit code by which field values are to be displayed. EDTCDE does not request any action by data base data management, but can be specified in the data base file description so that it can be duplicated into the display or printer file when this field is referenced at device file creation.	L/P-FLD	EDT

[1]Level:
L–Logical file
P–Physical file
FL–File level
RCD–Record level
FLD–Field level
KFLD–Key field level
S/O FLD–Select/omit field level

[2]Groups:
CHK–Validity checking keywords
EDT–Editing and field setting keywords
PTH–Access path keywords
REF–Naming and referencing keywords
S/O–Select/omit keywords
TXT–Text definition keywords

PHYSICAL AND LOGICAL FILE KEYWORDS (continued)

Keyword	Values	Function	Level[1]	Groups[2]
EDTWRD	('word')	This specifies an edit word that describes the form in which field values are to be displayed. EDTWRD does not request any action by data base data management, but can be specified in the data base file description so that it can be duplicated into the display or printer file when this field is referenced at display or printer file creation.	L/P-FLD	EDT
FORMAT	(data-base-file-name [.library-name])	A previously described record format is to be used.	L/P-RCD	REF
LIFO		Records with duplicate key values are processed in a last-in-first-out (LIFO) order.	L/P-FL	PTH
NOALTSEQ		The alternate collating sequence is not to be used for the key field.	L/P-KFLD	PTH
PFILE	(data-base-file-name [.library-name] [.32.])	This names the physical file(s) on which the logical file and logical record format is to be based.	L-RCD	REF
RANGE	(low high)	The field value must be within the specified range. RANGE is also used as a validity checking keyword, specified for retention in the data base file so it can be duplicated into a device file description through the reference function.	L/P-FLD L-S/O FLD	CHK S/O
REF	(data-base-file-name [.library-name] [format-name])	The system refers to the data base file for field specifications.	P-FL	REF

[1]**Level:**
L–Logical file
P–Physical file
FL–File level
RCD–Record level
FLD–Field level
KFLD–Key field level
S/O FLD–Select/omit field level

[2]**Groups:**
CHK–Validity checking keywords
EDT–Editing and field setting keywords
PTH–Access path keywords
REF–Naming and referencing keywords
S/O–Select/omit keywords
TXT–Text definition keywords

PHYSICAL AND LOGICAL FILE KEYWORDS (continued)

Keyword	Values	Function	Level[1]	Groups[2]
REFFLD	(referenced-field-name [.record-format-name] {*SRC / data-base-file-name [.library-name]})	The system refers to a field in a data base file for field specifications. Overrides REF keyword for this field.	P-FLD	REF
RENAME	(field-name)	This renames a physical file field name for a logical file record format.	L-FLD	REF
SIGNED		The sign of a field is considered when sequencing values.	L/P-KFLD	PTH
TEXT	('description')	This specifies a text description of a new record format or field in a new record format.	L/P-FLD L/P-RCD	TXT TXT
UNIQUE		Key values must be unique. (No duplicate key values allowed.)	L/P-FL	PTH
VALUES	(value-1 [value-2... [value-20]])	The field value must be one of the values specified. VALUES is also used as a validity checking keyword, specified for retention in the data base file so that it can be duplicated into a display file through the reference function.	L/P-FLD L-S/O FLD	CHK S/O
ZONE		Only the zone portion (high-order 4 bits) of each byte of the key field is used to build a key value.	L/P-KFLD	PTH

[1]Level:
L–Logical file
P–Physical file
FL–File level
RCD–Record level
FLD–Field level
KFLD–Key field level
S/O FLD–Select/omit field-level

[2]Groups:
CHK–Validity checking keywords
EDT–Editing and field setting keywords
PTH–Access path keywords
REF–Naming and referencing keywords
S/O–Select/omit keywords
TXT–Text definition keywords

PHYSICAL AND LOGICAL FILE KEYWORDS–GROUP LISTING

Access path definition keywords (PTH)

ABSVAL
ACCPTH
ALTSEQ
DESCEND
DIGIT
LIFO
NOALTSEQ
SIGNED
UNIQUE
ZONE

Editing and field setting keywords (EDT)

EDTCDE
EDTWRD

Naming and referencing keywords (REF)

CONCAT
FORMAT
PFILE
REF
REFFLD
RENAME

Select/omit keywords (S/O)

ALL
CMP
COMP
RANGE
VALUES

Text definition keywords (TXT)

COLHDG
TEXT

Validity checking keywords (CHK)

CHECK
CMP
COMP
RANGE
VALUES

PHYSICAL AND LOGICAL FILE KEYWORDS–LISTED BY LEVEL

Physical Files	Logical Files
File Level	**File Level**
[redacted]	ACCPTH
ALTSEQ	ALTSEQ
LIFO	LIFO
REF	[redacted]
UNIQUE	UNIQUE
Record Level	**Record Level**
FORMAT	FORMAT
[redacted]	PFILE (required for logical files)
TEXT	TEXT
Field Level	**Field Level**
CHECK(AB; ME; MF; M10; M11; VN)	CHECK(AB; ME; MF; M10; M11; VN)
CMP	CMP
COLHDG	COLHDG
COMP	COMP
[redacted]	CONCAT
EDTCDE	EDTCDE
EDTWRD	EDTWRD
RANGE	RANGE
REFFLD	RENAME
TEXT	TEXT
VALUES	VALUES
Key Field Level	**Key Field Level**
ABSVAL	ABSVAL
DESCEND	DESCEND
DIGIT	DIGIT
NOALTSEQ	NOALTSEQ
SIGNED	SIGNED
ZONE	ZONE
	Select/Omit Level
	ALL
	CMP
	COMP
	RANGE
	VALUES

Appendix C. Display File Keyword Summary

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
ALARM		An audible alarm is set on when the record is displayed.	RCD	O	DSP
ASSUME		The system assumes that the record is currently on the display when the file is opened.	RCD		MSC
AUTO		RA: See CHECK(ER) RAB: See CHECK(RB) RAZ: See CHECK(RZ)			
BLANKS	(response-indicator ['text'])	The response indicator is set on when a field containing all blanks is received from the device.	FLD		MSC
BLINK		When the record is displayed, the cursor blinks.	RCD	O	DSP
BLKFOLD		CPF folds the field at the last blank before the end of the line instead of at the actual end of the line.	FLD		DSP
CAnn	[(response-indicator ['text'])]	The command key specified by nn is a command *attention* key.	FL RCD	O O	KBD KBD
CFnn	[(response-indicator ['text'])]	The command key specified by nn is a command *function* key.	FL RCD	O O	KBD KBD
CHANGE	(response-indicator ['text'])	The response indicator is set on when data is changed in the field or in the record.	RCD FLD		CHK CHK
CHECK	(AB)	Allows blank input data to satisfy validity checking requirements for the field.	FL RCD FLD		CHK CHK CHK

[1]Level:
FL–File level
RCD–Record level
FLD–Field level

[2]Option Indicators:
O–Optional
R–Required
Blank–Invalid

[3]Groups:
CHK–Validity checking keywords
CTL–Processing control keywords
DSP–Display control keywords
EDT–Data editing/field setting keywords
KBD–Keyboard control keywords
MSC–Miscellaneous keywords
MSG–Message handling keywords
REF–Naming and referencing keywords
RTG–Routing keywords
SFL(C)–Subfile control record keywords
SFL(R)–Subfile record keywords
TXT–Text definition keywords

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
CHECK	(ER, FE, LC, RB, RZ)	On the 5250 work station, the following controls are placed on input keying: • End of record (ER) • Field exit check (FE) • Lowercase (LC) • Right-adjust, blank fill (RB) • Right-adjust, zero-fill (RZ)	FLD		KBD
CHECK	(ME, MF, M10, M11, VN)	A field value must meet one or more of the following check algorithms to be valid: • Mandatory enter (ME) or fill (MF) • Valid name (VN) • IBM modulus 10 (M10) or 11 (M11) self-check	FLD	(See note 1.)	CHK
CHECK	(RL)	The cursor moves from right to left within a field.	FL RCD FLD		DSP DSP DSP
CHECK	(RLTB)	The cursor advances between input-capable fields in a right-to-left, top-to-bottom manner.	FL		DSP
CHGINPDFT		The input-capable field is not to be underlined.	FL RCD FLD		DSP DSP DSP
CLEAR	[(response-indicator ['text'])]	Your program receives control when the Clear key is pressed.	FL RCD	O O (See note 2.)	KBD KBD
CMP		See COMP, (the preferred spelling).			
COMP	(EQ, NE, LT, NL, GT, NG, LE, GE)	A field value must meet the specified comparison test such as *equal to* to be valid.	FLD		CHK

[1]Level:
FL–File level
RCD–Record level
FLD–Field level

[2]Option Indicators:
O–Optional
R–Required
Blank–Invalid

[3]Groups:
CHK–Validity checking keywords
CTL–Processing control keywords
DSP–Display control keywords
EDT–Data editing/field setting keywords
KBD–Keyboard control keywords
MSC–Miscellaneous keywords
MSG–Message handling keywords
REF–Naming and referencing keywords
RTG–Routing keywords
SFL(C)–Subfile control record keywords
SFL(R)–Subfile record keywords
TXT–Text definition keywords

Notes:
1. Option indicators are valid for mandatory enter only.
2. A response indicator must be specified for this keyword to be used by RPG.

DISPLAY FILE KEYWORDS (continued)

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
CSRLOC	(field-name-1 field-name-2)	Your program determines the cursor location by setting the contents of field-name-1 and field-name-2.	RCD	O	DSP
DATE		The job date is displayed in the specified location.	FLD		EDT
DFT	('literal') | 'literal'	The field is initialized to the literal value. (This keyword is also used to define constant fields.)	FLD		EDT
DLTCHK		The field validity checking keywords are ignored (deleted) when referencing field specifications from a data base file.	FLD		REF
DLTEDT		The edit information is ignored (deleted) when referencing field specifications from a data base file.	FLD		REF
DSPATR	(attribute[attribute [attribute[...]]])	The following are display attributes for fields: • Column separator (CS) • High intensity (HI) • Underline (UL) • Blink field (BL) • Reverse image (RI) • Protect (PR) • Set modified data tag (MDT) • Nondisplay (ND) • Select by light pen (SP) • Position cursor (PC) • Operator identification (OID)	FLD	(See note.)	DSP

[1]Level:
FL–File level
RCD–Record level
FLD–Field level

[2]Option Indicators:
O–Optional
R–Required
Blank–Invalid

[3]Groups:
CHK–Validity checking keywords
CTL–Processing control keywords
DSP–Display control keywords
EDT–Data editing/field setting keywords
KBD–Keyboard control keywords
MSC–Miscellaneous keywords
MSG–Message handling keywords
REF–Naming and referencing keywords
RTG–Routing keywords
SFL(C)–Subfile control record keywords
SFL(R)–Subfile record keywords
TXT–Text definition keywords

Note: Option indicators not used for select by light pen or operator identification.

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
DSPSIZ	(*DSx [*DSy [*DSz]]) -or- (lines chars [condition-name-1] (lines chars [condition-name-2] (lines chars [condition-name-3]]])	This specifies the primary display size and any secondary display sizes. Valid entries for the parameter value are: *DS1 or 16 64 *DS2 or 12 80 *DS3 or 24 80 (x, y, or z = 1, 2, or 3 and x ≠ y ≠ z)	FL		DSP
DUP	[(response-indicator ['text'])]	Use of the Dup key is allowed.	FLD	O	KBD
EDTCDE	(edit-check-code [*|floating-currency-symbol])	This names the edit code by which field values are to be displayed.	FLD		EDT
EDTWRD	('edit-word')	This specifies an edit word that describes the form in which values are to be displayed.	FLD		EDT
ERASE	(record-name-1 [record-name-2... [record-name-20]])	One or more records are erased.	RCD	O	DSP
ERASEINP	{*MDTON *ALL}	*MDTON: Input-capable fields with MDTs set on are erased. *ALL: All input-capable fields are erased.	RCD	O	DSP
ERRMSG	('message' [response-indicator])	This specifies the message to be displayed on the message line.	FLD	O	MSG
ERRMSGID	(msgid msg-file [.library-name])	This specifies the message identifier of the message to be displayed on the messages line.	FLD	O	MSG
GETRETAIN		Retain all input data on the screen.	RCD		DSP
HELP	[(response-indicator ['text'])]	Your program receives control when the Help key is pressed.	FL RCD	O O	KBD KBD

[1]Level:
FL–File level
RCD–Record level
FLD–Field level

[2]Option Indicators:
O–Optional
R–Required
Blank–Invalid

[3]Groups:
CHK–Validity checking keywords
CTL–Processing control keywords
DSP–Display control keywords
EDT–Data editing/field setting keywords
KBD–Keyboard control keywords
MSC–Miscellaneous keywords
MSG–Message handling keywords
REF–Naming and referencing keywords
RTG–Routing keywords
SFL(C)–Subfile control record keywords
SFL(R)–Subfile record keywords
TXT–Text definition keywords

DISPLAY FILE KEYWORDS (continued)

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
HOME	[(response-indicator ['text'])]	Your program receives control when the Home key is pressed and the cursor is already in the home position.	FL RCD	O O	KBD KBD
INDTXT	(indicator 'indicator-usage-text')	Indicator text is to be associated with an indicator for high-level language program documentation.	FL RCD FLD		TXT TXT TXT
INZRCD		The record is written to the display before it is read.	RCD		DSP
KEEP		Erasing of the display is prevented when the file closes.	RCD		MSC
LOCK		The keyboard is not to be unlocked on a write operation.	RCD	O	KBD
LOGINP		The input record is copied (logged) to the job log.	RCD		MSC
LOGOUT		The output record is copied (logged) to the job log.	RCD	O	MSC
LOWER		See CHECK(LC) (the preferred spelling).			
MDTOFF	{*UNPR *ALL}	*UNPR: MDTs of fields without DSPATR(PR) are reset on an output operation. *ALL: MDTs of all fields are reset on an output operation.	RCD	O	DSP
MSGLOC	(line-number)	This specifies what line messages are to be displayed on.	FL	(See note.)	MSG
OPENPRT		The print file is to remain open until the display file is closed.	FL		MSC

[1]Level:
FL–File level
RCD–Record level
FLD–Field level

[2]Option Indicators:
O–Optional
R–Required
Blank–Invalid

[3]Groups:
CHK–Validity checking keywords
CTL–Processing control keywords
DSP–Display control keywords
EDT–Data editing/field setting keywords
KBD–Keyboard control keywords
MSC–Miscellaneous keywords
MSG–Message handling keywords
REF–Naming and referencing keywords
RTG–Routing keywords
SFL(C)–Subfile control record keywords
SFL(R)–Subfile record keywords
TXT–Text definition keywords

Note: Screen size condition names can be used as indicators (*DS1, *DS2, *DS3).

DISPLAY FILE KEYWORDS (continued)

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
OVERLAY		The entire display is not erased before this record is displayed.	RCD	O	DSP
OVRATR		Display attributes of this field are changed on an output operation.	FLD	O	DSP
OVRDTA		Data contents of this field are changed on an output operation.	FLD	O	DSP
PASSRCD	(record-format)	This specifies the record format to be used when unformatted data is passed.	FL		CTL
PRINT	[(response-indicator ['text']) |(printer-file-name [.library-name])]	The Print key can be used to: • Print to the associated device. • Print to the specified printer file. • Pass control to your program.	FL	O (See note.)	KBD
PROTECT		All input-capable field on the screens are changed to output-only fields.	RCD	O	DSP
PUTOVR		Permits the use of OVRATR and OVRDTA keywords.	RCD	O	DSP
PUTRETAIN		A record (at the record format level) or field (at the field level) on the display is retained.	RCD FLD	O O	DSP DSP
RANGE	(low high)	The field value must be within the range specified.	FLD		CHK
REF	(data-base-file-name [.library-name] [record-format-name])	The system refers to the data base file for field specifications for this display file.	FL		REF

[1]**Level:**
FL–File level
RCD–Record level
FLD–Field level

[2]**Option Indicators:**
O–Optional
R–Required
Blank–Invalid

[3]**Groups:**
CHK–Validity checking keywords
CTL–Processing control keywords
DSP–Display control keywords
EDT–Data editing/field setting keywords
KBD–Keyboard control keywords
MSC–Miscellaneous keywords
MSG–Message handling keywords
REF–Naming and referencing keywords
RTG–Routing keywords
SFL(C)–Subfile control record keywords
SFL(R)–Subfile record keywords
TXT–Text definition keywords

Note: A response indicator must be specified for this keyword to be used by COBOL.

DISPLAY FILE KEYWORDS (continued)

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
REFFLD	(referenced-field-name [.record-format-name] {*SRC / data-base-file-name [.library-name]})	The system refers to the data base file (other than the file specified in the REF keyword) for field specifications for this display file.	FLD		REF
ROLLDOWN	[(response-indicator ['text'])]	Your program receives control when the Roll Down key is pressed.	FL RCD	O O	KBD KBD
ROLLUP	[(response-indicator ['text'])]	Your program receives control when the Roll Up key is pressed.	FL RCD	O O	KBD KBD
RTGAID	(starting-routing-field-location)	The AID byte (command key indication) is placed into the routing data as a 2-byte identifier.	RCD		RTG
RTGCON	(starting-routing-field-location 'constant')	The specified literal is placed into the routing data at the specified location.	RCD		RTG
RTGDEV	(starting-routing-field-location)	The 10-character device name is placed into the routing data.	RCD		RTG
RTGDEVCLS	(starting-routing-field-location)	The 2-character device class is placed into the routing data.	RCD		RTG
RTGFIRST	(starting-routing-field-location entry-length)	The length and location of the first field received is placed in the routing data.	RCD		RTG
RTGFLD	(starting-routing-field-location)	The field data is placed in the routing data.	FLD		RTG
RTGFMT	(starting-routing-field-location)	The record format name is placed into the routing data.	RCD		RTG
RTGPOS	(starting-routing-field-location starting-data-position ending-data-position)	Data received from a display device is placed into the routing data.	RCD		RTG

[1]Level:
FL–File level
RCD–Record level
FLD–Field level

[2]Option Indicators:
O–Optional
R–Required
Blank–Invalid

[3]Groups:
CHK–Validity checking keywords
CTL–Processing control keywords
DSP–Display control keywords
EDT–Data editing/field setting keywords
KBD–Keyboard control keywords
MSC–Miscellaneous keywords
MSG–Message handling keywords
REF–Naming and referencing keywords
RTG–Routing keywords
SFL(C)–Subfile control record keywords
SFL(R)–Subfile record keywords
TXT–Text definition keywords

DISPLAY FILE KEYWORDS (continued)

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
RTNDTA		CPF returns the same input data as on the last input operation.	RCD		CTL
SETOF	(response-indicator ['text'])	The specified response indicator is set off on an input operation.	RCD		MSC
SETOFF		See SETOF (the preferred spelling).			
SFL		The record format is a subfile record format.	RCD		SFL(R)
SFLCLR		This allows the clearing of a subfile (not the same as the deletion of a subfile).	RCD	R	SFL(C)
SFLCTL	(record-format-name)	The record format is a subfile control record format.	RCD		SFL(C)
SFLDLT		This allows the deletion of a subfile.	RCD	R	SFL(C)
SFLDROP	(CAnn/CFnn)	This assigns a command key to be used to fold or truncate records of a subfile.	RCD		SFL(C)
SFLDSP		This allows the display of the subfile.	RCD	O	SFL(C)
SFLDSPCTL		This allows the display of the subfile control record.	RCD	O	SFL(C)
SFLEND		This allows the display of a plus sign when there are more subfile records than fit on one page.	RCD	R	SFL(C)
SFLENTER	(CAnn/CFnn)	This changes the Enter/Rec Adv key to a Roll Up key and assigns the specified command key to be Enter/Rec Adv key.	RCD		SFL(C)
SFLINZ		This allows all records within the subfile to be initialized with one output operation.	RCD	O	SFL(C)

[1]Level:
FL–File level
RCD–Record level
FLD–Field level

[2]Option Indicators:
O–Optional
R–Required
Blank–Invalid

[3]Groups:
CHK–Validity checking keywords
CTL–Processing control keywords
DSP–Display control keywords
EDT–Data editing/field setting keywords
KBD–Keyboard control keywords
MSC–Miscellaneous keywords
MSG–Message handling keywords
REF–Naming and referencing keywords
RTG–Routing keywords
SFL(C)–Subfile control record keywords
SFL(R)–Subfile record keywords
TXT–Text definition keywords

DISPLAY FILE KEYWORDS (continued)

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
SFLLIN	(spaces)	This specifies the number of spaces between columns of subfile records.	RCD	(See note.)	SFL(C)
SFLMSG	('message' [response-indicator])	This specifies the subfile-related message to be displayed on the message line.	RCD	O	SFL(C)
SFLMSGID	(msgid msg-file [.library-name])	This specifies the message identifier of the subfile-related message to be displayed on the message line.	RCD	O	SFL(C)
SFLMSGKEY		This field contains the message identifier (message key) of the message on a program message queue to be placed in a message subfile.	FLD		SFL(R)
SFLMSGRCD	(line-number)	The subfile is a message subfile and contains messages from a program message queue. SFLMSGRCD is valid on the subfile record format only.	RCD	(See note.)	SFL(R)
SFLNXTCHG		Your program can force the work station user to rekey input fields correctly.	RCD	O	SFL(R)

[1]**Level:**
FL–File level
RCD–Record level
FLD–Field level

[2]**Option Indicators:**
O–Optional
R–Required
Blank–Invalid

[3]**Groups:**
CHK–Validity checking keywords
CTL–Processing control keywords
DSP–Display control keywords
EDT–Data editing/field setting keywords
KBD–Keyboard control keywords
MSC–Miscellaneous keywords
MSG–Message handling keywords
REF–Naming and referencing keywords
RTG–Routing keywords
SFL(C)–Subfile control record keywords
SFL(R)–Subfile record keywords
TXT–Text definition keywords

Note: Screen size condition names can be used as indicators (*DS1, *DS2, *DS3).

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
SFLPAG	(number-to-be-displayed)	This specifies the number of subfile records that can be displayed at one time. If field selection is used in the subfile, this represents the number of display lines in the page.	RCD	(See note.)	SFL(C)
SFLPGMQ		This field contains the name of the program message queue containing messages to be placed in a message subfile.	FLD		SFL(C), SFL(R)
SFLRCDNBR	[(CURSOR)]	A page of a subfile is displayed according to a relative record number.	FLD		SFL(C)
SFLRNA		This keyword enables a subfile to be initialized with no active records.	RCD		SFL(C)
SFLROLVAL		The specified field and the appropriate Roll key are used to roll the subfile records that are displayed.	FLD		SFL(C)
SFLSIZ	(number-of-records-in-subfile)	This specifies the number of records in a subfile.	RCD	(See note.)	SFL(C)
SLNO	(nn|*VAR)	Line numbers for all fields in this record format are increased by nn or by the value of the field *VAR as set by your program before the record is displayed.	RCD		DSP
TEXT	('description')	This is a text description for a record format or field.	RCD FLD		TXT TXT
TIME		The system time is displayed in the specified location.	FLD		EDT

[1]Level:
FL–File level
RCD–Record level
FLD–Field level

[2]Option Indicators:
O–Optional
R–Required
Blank–Invalid

[3]Groups:
CHK–Validity checking keywords
CTL–Processing control keywords
DSP–Display control keywords
EDT–Data editing/field setting keywords
KBD–Keyboard control keywords
MSC–Miscellaneous keywords
MSG–Message handling keywords
REF–Naming and referencing keywords
RTG–Routing keywords
SFL(C)–Subfile control record keywords
SFL(R)–Subfile record keywords
TXT–Text definition keywords

Note: Screen size condition names can be used as indicators (*DS1, *DS2, *DS3).

DISPLAY FILE KEYWORDS (continued)

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
UNLOCK	[*ERASE|*MDTOFF]| [*ERASE *MDTOFF]| *MDTOFF *ERASE]	The keyboard is unlocked after an input operation so that the next record can be entered before the next output operation.	RCD		KBD
USRDFN		The data is a user-defined data stream.	RCD		DSP
VALUES	(value-1 [value-2... [value-20]])	The field value must be one of the values specified.	FLD		CHK
VLDCMDKEY	(response-indicator ['text'])	The response indicator is set on if a valid command key (a key associated with a keyword) is pressed.	FL RCD		KBD KBD

[1]**Level:**
FL–File level
RCD–Record level
FLD–Field level

[2]**Option Indicators:**
O–Optional
R–Required
Blank–Invalid

[3]**Groups:**
CHK–Validity checking keywords
CTL–Processing control keywords
DSP–Display control keywords
EDT–Data editing/field setting keywords
KBD–Keyboard control keywords
MSC–Miscellaneous keywords
MSG–Message handling keywords
REF–Naming and referencing keywords
RTG–Routing keywords
SFL(C)–Subfile control record keywords
SFL(R)–Subfile record keywords
TXT–Text definition keywords

DISPLAY FILE KEYWORDS–GROUP LISTING

Data editing/field setting keywords (EDT)

DATE
DFT
EDTCDE
EDTWRD
TIME

Display control keywords (DSP)

ALARM
BLINK
BLKFOLD
CHECK(RL or RLTB only)
CHGINPDFT
CLRL
CSRLOC
DSPATR
DSPSIZ
ERASE
ERASEINP
GETRETAIN
INZRCD
MDTOFF
OVERLAY
OVRATR
OVRDTA
PROTECT
PUTOVR
PUTRETAIN
SLNO
USRDFN

Keyboard control keywords (KBD)

CAnn
CFnn
CHECK(ER, FE, LC, RB, RZ)
CLEAR
DUP
HELP
HOME
LOCK
LOWER
PRINT
ROLLDOWN
ROLLUP
UNLOCK
VLDCMDKEY

Message handling keywords (MSG)

ERRMSG
ERRMSGID
MSGLOC

Miscellaneous keywords (MSC)

ASSUME
BLANKS
KEEP
LOGINP
LOGOUT
OPENPRT
SETOF
SETOFF

Naming and referencing keywords (REF)

DLTCHK
DLTEDT
REF
REFFLD

Processing control keywords (CTL)

PASSRCD
RTNDTA

Routing keywords (RTG)

RTGAID
RTGCON
RTGDEV
RTGDEVCLS
RTGFIRST
RTGFLD
RTGFMT
RTGPOS

Subfile control record keywords (SFL(C))

SFLCLR
SFLCTL
SFLDLT
SFLDROP
SFLDSP
SFLDSPCTL
SFLEND
SFLENTER
SFLINZ
SFLLIN
SFLMSG
SFLMSGID
SFLPAG
SFLPGMQ
SFLRCDNBR
SFLRNA
SFLROLVAL
SFLSIZ

Subfile record keywords (SFL(R))

SFL
SLFMSGKEY
SFLMSGRCD
SFLNXTCHG
SFLPGMQ

Text definition keywords (TXT)

INDTXT
TEXT

Validity checking keywords (CHK)

AUTO
CHANGE
CHECK(AB, ME, MF, M10, M11, VN)
CMP
COMP
RANGE
VALUES

DISPLAY FILE KEYWORDS–LISTED BY LEVEL

File Level

CAnn
CFnn
CHECK(AB)
CHECK(RL) and CHECK(RLTB)
CHGINPDFT
CLEAR
DSPSIZ
HELP
HOME
INDTXT
MSGLOC
OPENPRT
PASSRCD
PRINT
REF
ROLLDOWN
ROLLUP
VLDCMDKEY

Record Level

ALARM
ASSUME
BLANKS
BLINK
CAnn
CFnn
CHANGE
CHGINPDFT
CLEAR
CHECK(AB)
CHECK(RL)
CLRL
CSRLOC
ERASE
ERASEINP
GETRETAIN
HELP
HOME
INDTXT
INZRCD
KEEP
LOCK
LOGINP
LOGOUT
MDTOFF
OVERLAY
PROTECT
PUTOVR
PUTRETAIN
ROLLDOWN
ROLLUP
RTGAID
RTGCON
RTGDEV
RTGDEVCLS
RTGFIRST
RTGFMT
RTGPOS
RTNDTA
SETOF
SETOFF
SFL
SFLCLR
SFLCTL
SFLDLT
SFLDROP
SFLDSP
SFLDSPCTL
SFLEND
SFLENTER
SFLINZ
SFLLIN
SFLMSG
SFLMSGID
SFLMSGRCD
SFLNXTCHG
SFLPAG
SFLRNA
SFLSIZ
SLNO
TEXT
UNLOCK
USRDFN
VLDCMDKEY

Field Level

AUTO
BLKFOLD
CHANGE
CHECK(AB; ER; FE; LC; ME; MF;
M10; M11; RB; RZ; VN)
CHECK(RL)
CHGINPDFT
CMP
COMP
DATE
DFT
DLTCHK
DLTEDT
DSPATR
DUP
EDTCDE
EDTWRD
ERRMSG
ERRMSGID
INDTXT
LOWER
OVRATR
OVRDTA
PUTRETAIN
RANGE
REFFLD
RTGFLD
SFLMSGKEY
SFLPGMQ
SFLRCDNBR
SFLROLVAL
TEXT
TIME
VALUES

Display File Keywords–Combined Chart

Display File																		
Keyword	**Level**					**Usage**					**Keyboard Shifts**							
	File	**Record**	**Field**	**Option Indicators**	**Response Indicators**	**Input**	**Output**	**Both**	**Hidden**	**Message**	**Alpha Shift**	**Num Shift**	**Signed Num**	**Alpha Only**	**Num Only**	**Katakana**	**Inhibit**
ALARM		X		O													
ASSUME		X															
AUTO(RA)			X	O		X		X			X	X	X	X	X	X	X
AUTO(RAB)			X			X		X			X	X	X	X	X	X	X
AUTO(RAZ)			X			X		X			X	X	X	X	X	X	X
BLANKS			X			X		X			X	X	X	X	X	X	X
BLINK		X		O													
BLKFOLD		X					X				X	X	X	X	X	X	X
CA01-CA24	X	X		O	O												
CF01-CF24	X	X		O	O												
CHANGE		X	X		R	X		X			X	X	X	X	X	X	X
CHECK(AB)	X	X	X			X		X			X	X	X	X	X	X	X
CHECK(ER)			X	O		X		X			X	X	X	X	X	X	X
CHECK(FE)			X			X		X			X	X	X	X	X	X	X
CHECK(LC)			X			X		X			X	X		X		X	X
CHECK(ME)			X	O		X		X			X	X	X	X	X	X	X
CHECK(MF)			X			X		X			X	X	X	X	X	X	X
CHECK(M10)			X			X		X			X	X	X	X	X	X	X
CHECK(M11)			X			X		X			X	X	X	X	X	X	X
CHECK(RB)			X			X		X			X	X	X	X	X	X	X
CHECK(RL)	X	X	X			X		X			X						
CHECK(RLTB)	X																
CHECK(RZ)			X			X		X			X	X	X	X	X	X	X
CHECK(VN)			X			X		X			X	X		X		X	X
CLRL		X															
CHGINPDFT	X	X	X			X	X				X	X	X	X	X	X	X
CMP			X			X		X			X	X	X	X	X	X	X
COMP			X			X		X			X	X	X	X	X	X	X

X = Allowed
O = Optional
R = Required
Blank = Not allowed

Display File Keywords–Combined Chart (continued)

Display File																		
Keyword	Level					Usage					Keyboard Shifts							
	File	Record	Field	Option Indicators	Response Indicators	Input	Output	Both	Hidden	Message	Alpha Shift	Num Shift	Signed Num	Alpha Only	Num Only	Katakana	Inhibit
CSRLOC		X		O													
DATE			X				X[1]										
DFT			X			X	X	X			X	X	X	X	X	X	X
DLTCHK			X			X		X			X	X	X	X	X	X	X
DLTEDT			X				X	X			X	X	X	X	X	X	
DSPATR(BL)			X	O		X	X	X			X	X	X	X	X	X	X
DSPATR(CS)			X	O		X	X	X			X	X	X	X	X	X	X
DSPATR(HI)			X	O		X	X	X			X	X	X	X	X	X	X
DSPATR(MDT)			X	O		X		X			X	X	X	X	X	X	X
DSPATR(ND)			X	O		X	X	X			X	X	X	X	X	X	X
DSPATR(OID)			X			X		X			X	X	X	X	X	X	X
DSPATR(PC)			X	O		X	X	X			X	X	X	X	X	X	X
DSPATR(PR)			X	O		X		X			X	X	X	X	X	X	X
DSPATR(RI)			X	O		X	X	X			X	X	X	X	X	X	X
DSPATR(SP)			X			X		X			X	X	X	X	X	X	X
DSPATR(UL)			X	O		X	X	X			X	X	X	X	X	X	X
DSPSIZ	X																
DUP			X	O	O	X		X			X	X	X	X	X	X	X
EDTCDE			X				X	X							X		
EDTWRD			X				X	X							X		
ERASE		X		O													
ERASEINP		X		O													
ERRMSG			X	O	O	X	X	X			X	X	X	X	X	X	X
ERRMSGID			X	O		X	X	X			X	X	X	X	X	X	X
GETRETAIN		X															
HELP	X	X		O	O												
HOME	X	X		O	O												
INDTXT	X	X	X		R	X	X	X	X	X	X	X	X	X	X	X	X
INZRCD		X															
KEEP		X															
LOCK		X		O													
LOGINP		X															
LOGOUT		X		O													
LOWER			X			X		X			X	X		X		X	X
MDTOFF		X		O													

X = Allowed
X[1] = Constant field only
O = Optional
R = Required
Blank = Not allowed

Display File Keywords–Combined Chart (continued)

Display File																		
Keyword	**Level**					**Usage**					**Keyboard Shifts**						
	File	Record	Field	Option Indicators	Response Indicators	Input	Output	Both	Hidden	Message	Alpha Shift	Num Shift	Signed Num	Alpha Only	Num Only	Katakana	Inhibit
MSGLOC	X			*													
OPENPRT	X																
OVERLAY		X		O													
OVRATR			X	O		X	X	X			X	X	X	X	X	X	X
OVRDTA			X	O		X	X	X		X	X	X	X	X	X	X	X
PASSRCD	X																
PRINT	X			O	O												
PROTECT		X		O													
PUTRETAIN		X	X	O		X	X	X			X	X	X	X	X	X	X
PUTOVR		X		O													
RANGE			X			X		X			X	X	X	X	X	X	X
REF	X																
REFFLD			X			X	X	X	X	X	X	X	X	X	X	X	X
ROLLDOWN	X	X		O	O												
ROLLUP	X	X		O	O												
RTGAID		X															
RTGCON		X															
RTGDEV		X															
RTGDEVCLS		X															
RTGFIRST		X															
RTGFLD			X			X		X			X	X	X	X	X	X	X
RTGFMT		X															
RTGPOS		X															
RTNDTA		X															
SETOF		X			R												
SETOFF		X			R												
SFL		X															
SFLCLR		X		R													
SFLCTL		X															
SFLDLT		X		R													
SFLDROP		X															
SFLDSP		X		O													
SFLDSPCTL		X		O													
SFLEND		X		R													
SFLENTER		X															

X = Allowed
O = Optional
R = Required
Blank = Not allowed
* = Screen size condition names allowed

Display File Keywords–Combined Chart (continued)

Display File																		
Keyword	**Level**					**Usage**					**Keyboard Shifts**							
	File	Record	Field	Option Indicators	Response Indicators	Input	Output	Both	Hidden	Message	Alpha Shift	Num Shift	Signed Num	Alpha Only	Num Only	Katakana	Inhibit
SFLINZ		X		O													
SFLLIN		X		*													
SFLMSG		X		O	O												
SFLMSGID		X		O													
SFLMSGKEY			X														
SFLMSGRCD		X		*													
SFLNXTCHG		X		O													
SFLPAG		X		*													
SFLPGMQ			X														
SFLRCDNBR			X				X	X	X				X				
SFLRNA		X															
SFLROLVAL			X			X		X					X				
SFLSIZ		X		*													
SLNO		X															
TEXT		X	X			X	X	X	X	X	X	X	X	X	X	X	X
TIME			X				X[1]										
UNLOCK		X															
USRDFN		X															
VALUES			X			X		X			X	X	X	X	X	X	X
VLDCMDKEY	X	X			R												

X = Allowed
X[1] = Constant field only
O = Optional
R = Required
Blank = Not allowed
* = Screen size condition names allowed

Appendix D. Printer File Keyword Summary

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
BLKFOLD		Records are folded at the last blank before the end of the line instead of folded at the actual end of the line.	FLD		PRT
DATE		The job date is printed in the specified location.	FLD		EDT
DFT	('literal') | 'literal'	A field is initialized to a constant value.	FLD		EDT
DLTEDT		The edit information is ignored when referencing field specifications.	FLD		REF
EDTCDE	(edit-code [* | floating-currency -symbol])	This names the edit code by which field values are to be printed.	FLD		EDT
EDTWRD	('edit-word')	This specifies an edit word that describes the form in which values are to be printed.	FLD		EDT
INDTXT	(indicator 'indicator-usage -text')	Indicator text is to be associated with an indicator for high-level language program documentation.	FL RCD FLD		TXT TXT TXT
PAGNBR		A system-supplied page number is printed in this field (a 4-digit zoned decimal field).	FLD	O	PRT

[1]Level:
FL–File level
RCD–Record level
FLD–Field level

[2]Option Indicators:
O–Optional
R–Required
Blank–Invalid

[3]Groups:
EDT–Editing and field setting keywords
PRT–Printer control
REF–Naming and referencing keywords
TXT–Text definition keywords

PRINTER FILE KEYWORDS (continued)

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
REF	(data-base-file-name [.library-name] [format-name])	The system refers to field specifications from a referenced data base file.	FL		REF
REFFLD	(referenced-field-name [.record-format-name] {*SRC {data-base-file-name [.library-name]})	The system refers to field specifications from a data base file (other than the file specified in the REF keyword).	FLD		REF
SKIPA	(skip-after-line-number)	This specifies a line to skip to after printing a line (file level), after printing an entire record (record format level), or after printing the line containing the field associated with this keyword (field level).	FL RCD FLD	R O O	PRT PRT PRT
SKIPB	(skip-before-line-number)	This specifies the line to skip to before printing the next line of output (file level), or before printing the lines associated with a record (record format level), or before printing the line containing the field associated with this keyword (field level).	FL RCD FLD	R O O	PRT PRT PRT

[1]**Level:**
FL–File level
RCD–Record level
FLD–Field level

[2]**Option Indicators:**
O–Optional
R–Required
Blank–Invalid

[3]**Groups:**
EDT–Editing and field setting keywords
PRT–Printer control
REF–Naming and referencing keywords
TXT–Text definition keywords

PRINTER FILE KEYWORDS (continued)

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
SPACEA	(space-after-value)	This specifies the number of lines to space after printing an entire record (record format level) or to space after printing the line containing the field associated with this keyword (field level).	RCD FLD	O O	PRT PRT
SPACEB	(space-before-value)	This specifies the number of lines to space before printing the lines associated with a record (record format level) or before printing the line containing the field associated with this keyword (field level).	RCD FLD	O O	PRT PRT
TEXT	('description')	This is text description of the record format (record level) or descriptions of the field (field level).	RCD FLD		TXT TXT
TIME		The system time is printed in the specified location.	FLD		EDT
UNDERLINE		The field is underlined.	FLD	O	PRT

[1]**Level:**
FL–File level
RCD–Record level
FLC–Field level

[2]**Option Indicators:**
O–Optional
R–Required
Blank–Invalid

[3]**Groups:**
EDT–Editing and field setting keywords
PRT–Printer control
REF–Naming and referencing keywords
TXT–Text definition keywords

PRINTER FILE KEYWORDS–GROUP LISTING

Editing and field setting keywords (EDT)

DATE
DFT
EDTCDE
EDTWRD
TIME

Naming and referencing keywords (REF)

DLTEDT
REF
REFFLD

Printer control keywords (PRT)

BLKFOLD
PAGNBR
SKIPA
SKIPB
SPACEA
SPACEB
UNDERLINE

Text definition keywords (TXT)

INDTXT
TEXT

PRINTER FILE KEYWORDS–LISTED BY LEVEL

File Level

INDTXT
REF
SKIPA
SKIPB

Record Level

INDTXT
SKIPA
SKIPB
SPACEA
SPACEB
TEXT

Field Level

BLKFOLD
DATE
DFT
DLTEDT
EDTCDE
EDTWRD
INDTXT
PAGNBR
REFFLD
SKIPA
SKIPB
SPACEA
SPACEB
TEXT
TIME
UNDERLINE

Appendix E. Communications File and BSC File Keyword Summary

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
ALWWRT		The record currently being written ends a transmission; the program is going to receive state.	FL RCD	O O	CTL CTL
ENDGRP		Indicates the end of a user-defined group of records. Your program remains in a send state.	FL RCD	O	CTL
FAIL		Discard the records being sent in the current transmission.	FL RCD	O O	CTL CTL
INDTXT	(indicator 'indicator-usage-text')	Indicator text is to be associated with an indicator for high-level language program documentation.	FL RCD		TXT TXT
RCVENDGRP	(response-indicator ['text'])	Indicates the end of a user-defined group of records sent to the program.	FL RCD		MSC MSC
REF	(data-base-file-name [.library-name] [record-format-name])	The system refers to field specifications from a referenced data base file.	FL		REF
REFFLD	(referenced-field-name [.record-format-name] {*SRC / [data-base-file-name [.library-name]]})	The system refers to field specifications from a data base file (other than the file specified in the REF keyword).	FLD		REF
RQSWRT		Requests permission from the host system so the user can send data.	FL RCD	O O	MSC MSC
RTGDEV	(starting-routing-field-location)	The 10-character device name is placed into the routing data.	RCD		RTG

[1]**Level:**
FL–File level
RCD–Record level
FLD–Field level

[2]**Option Indicators:**
O–Optional
R–Required
Blank–Invalid

[3]**Groups:**
CTL–Processing control keywords
MSC–Miscellaneous keywords
REF–Naming and referencing keywords
RTG–Routing keywords
TXT–Text definition keywords

COMMUNICATIONS FILE AND BSC FILE KEYWORDS (continued)

Keyword	Values	Function	Level[1]	Option Indicators[2]	Groups[3]
RTGDEVCLS	(starting-routing-field-location)	The 2-character device class is placed into the routing data.	RCD		RTG
RTGFMT	(starting-routing-field-location)	The format name is placed into the routing data.	RCD		RTG
RTGPOS	(starting-routing-field-location starting-data-position ending-data-position)	Data received from a device is placed into the routing data.	RCD		RTG
TEXT	('description')	This is a text description for a record format or a field.	RCD FLD		TXT TXT
TRNRND	(response-indicator ['text'])	Indicates to the program that it is now in the send state and has permission to write.	FL RCD		CTL CTL

[1]Level:
FL–File level
RCD–Record level
FLD–Field level

[2]Option Indicators:
O–Optional
R–Required
Blank–Invalid

[3]Groups:
CTL–Processing control keywords
MSC–Miscellaneous keywords
REF–Naming and referencing keywords
RTG–Routing keywords
TXT–Text definition keywords

COMMUNICATIONS FILE AND BSC FILE KEYWORDS–GROUP LISTING

Miscellaneous keywords (MSC)

RCVENDGRP
RQSWRT

Naming and referencing keywords (REF)

REF
REFFLD

Processing control keywords (CTL)

ALWWRT
ENDGRP
FAIL
TRNRND

Routing keywords (RTG)

RTGDEV
RTGDEVCLS
RTGFMT
RTGPOS

Text definition keywords (TXT)

INDTXT
TEXT

COMMUNICATIONS FILE AND BSC FILE KEYWORDS–LISTED BY LEVEL

File Level

ALWWRT
ENDGRP
FAIL
INDTXT
REF
RCVENDGRP
RQSWRT
TRNRND

Record Level

ALWWRT
ENDGRP
FAIL
INDTXT
RCVENDGRP
RQSWRT
RTGDEV
RTGDEVCLS
RTGFMT
RTGPOS
TEXT
TRNRND

Field Level

REFFLD
TEXT

Communications File and BSC File Keywords–Combined Chart

Keywords	Level			Option Indicators	Response Indicators	Send State	Receive State
	File	Record	Field				
ALWWRT	X	X		O		Put	
ENDGRP	X	X		O		Put	
FAIL	X	X		O		Put[1]	Put[1]
RCVENDGRP	X	X			R		Get
RQSWRT	X	X		O			Put[2]
TRNRND	X	X			R		Get

X = Allowed
O = Optional
R = Required
Blank = Not allowed
Put = Can be selected only for output operations
Get = Can be selected only for input operations

Notes:
1. If FAIL is in effect, ALWWRT, ENDGRP, and RQSWRT are ignored.
2. If RQSWRT is in effect and FAIL is not in effect, ALWWRT and ENDGRP are ignored.

Appendix F. Examples

This appendix describes sample data description specifications for each type of file covered in this manual. The samples chosen could be made to work with appropriate high-level language programs. Two such programs are shown in skeletal form to illustrate the use of externally described data. There are four sections in this appendix:

- Data base files
  - A field reference file (a physical file used for reference, not data storage)
  - Two physical files
  - Two logical files
- Device files
  - A simple menu
  - An inquiry display with two record formats
  - Twelve subfile examples
  - A printer file
  - A communications file or a BSC file
- Externally described data in RPG and COBOL
  - File specifications in RPG
  - The data division in COBOL
- A sample compiler listing (output from a create file command) and the IBM Data Description Specifications Debugging Template (reduced)

The following list describes each of the examples in this appendix:

Data Base File Examples

Example 1. Field Reference File

- Important keywords: TEXT, COLHDG, EDTCDE(Z), REFFLD
- One record format
- Defines all fields used in an application
- References fields only within the field reference file itself

Example 2. Physical File with a New Record Format

- Important keywords: REF
- One record format
- Names of all fields in the record format are specified
- References fields in a reference file (REF keyword)
- Keyed sequence access path

Example 3. Physical File Using FORMAT and a Keyed Sequence Access Path

- Important keywords: UNIQUE, FORMAT
- Composite key

Example 4. Logical File Specifying New Keys

- Important keywords: PFILE
- New field specifications
- Provides two record formats
- Each record format uses a key different from the based-on physical file

Example 5. Logical File Specifying a New Record Format

- Important keywords: UNIQUE
- Specifies a record format different from the based-on physical file

Device File Examples

Example 6. Simple Menu

- Important keywords: BLINK, VALUES
- Constant fields
- Numeric-only, input-only fields

Example 7. Inquiry Display with Two Record Formats

- Important keywords: PRINT, CAnn, CHECK, OVERLAY, EDTCDE(2 $), DSPATR(UL), DSPATR(HI BL), EDTCDE(Y), ERRMSG
- Use of +n to specify position

Example 8. Work Station Display Showing Subfile Operations

- Important keywords: CAnn, DSPATR(HI), SETOF, ERRMSG, SFL, SFLCTL, SFLCLR, SFLDSP, SFLDSPCTL, SFLSIZ, SFLPAG
- Option indicators (specified with and without N)
- Four-step sequence of operations explained

Example 9. Subfile with SFLPAG Value Equal to SFLSIZ Value

- Important keywords: SFLSIZ, SFLPAG, SFLDSP, SFLDSPCTL, SFLCLR, ROLLUP, ROLLDOWN
- One column of subfile records
- Constant fields in subfile control record format used as headings for columns of fields in subfile records

Example 10. Subfile for Update without Validity Checking

- Important keywords: CHECK(LC), CHECK(RB), UNLOCK(*ERASE *MDTOFF)
- No validity checking keywords specified on subfile record format
- Permits rapid data entry to a transaction file that is run later to update existing records in the data base

Example 11. Subfile for Update with Validity Checking

- Important keywords: SFLNXTCHG, COMP, DSPATR, DATE, TIME
- Validity checking keywords and SFLNXTCHG in the subfile record format permit the program to ensure accurate data entry

Example 12. Subfile with SFLINZ and SFLRNA Keywords

- Important keywords: SFLNXTCHG, SFLINZ, SFLRNA
- Display shows the subfile just after the program has initialized it; the program can write records to the subfile or the work station user can modify them

Example 13. Subfile with SFLDROP Keyword

- Important keywords: SFLDROP
- Displays show the subfile in its two forms: folded and truncated

Example 14. Subfile with Variable Length Records (Field Selection)

- Important keywords: SFLPAG, SFLSIZ
- The display shows a subfile in which some records are displayed with two fields and some are displayed with three fields

Example 15. Subfile with Rolling by CPF

- Important keywords: SFLPAG, SFLSIZ
- SFLSIZ value is larger than SFLPAG value
- Subfile is rolled by the SFLPAG value

Example 16. Subfile with Rolling by CPF (SFLRCDNBR and SFLROLVAL Keywords)

- Important keywords: SFLRCDNBR, SFLROLVAL
- Subfile is rolled by a certain number of records.

Example 17. Horizontal Subfile That Is Also Displayable on Two Display Sizes

- Important keywords: SFLLIN, DSPSIZ
- Subfile records appear in two columns (SFLLIN keyword)
- Subfile can be displayed on two display sizes (DSPSIZ keyword)

Example 18. Message Subfile

- Important keywords: SFLMSGRCD, SFLMSGKEY, SFLMSGQ
- Records in the subfile are first-level text of messages in a message file

Example 19. Display File with Three Subfiles

- Important keywords: OVERLAY, SFLDLT
- The first display shows a horizontal subfile and a vertical subfile; the second display shows the horizontal subfile and a message subfile

Example 20. Printer File

- Important keywords: SKIPB, EDTCDE(Y), PAGNBR, EDTCDE(Z), SPACEA, UNDERLINE
- This printer file uses space and skip keywords instead of line numbers

Example 21. Communications File or BSC File

- Important keywords: ALWWRT, RQSWRT, FAIL, RCVENDGRP, TRNRND
- This file could be either a communications file (SNA protocol) or a BSC file, depending on the create file command used

Externally Described Data in RPG and COBOL

Example 22. File Specifications in RPG

- For data base files (DISK) and display files (WORKSTN) with one subfile

Example 23. Data Division in COBOL

- For a data base file

Example 1. Field Reference File

The following field reference file (MLGREFP) describes all fields used by any program in the application. The other files reference the fields in this file.

The following numbered items apply to Figure F-1:

1. Like all physical files, a field reference file has only one record format. The R in position 17 specifies that MLGREFR is the record format name.

2. The Rs in position 29 and REFFLD in positions 45 through 80 specify that the fields ADDR and CITY are to have the same attributes as NAME.

3. Specifying COLHDG for ADDR and CITY overrides the COLHDG attribute for NAME, which otherwise would have been in effect.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/060*
Printed in U.S.A.

File | Programmer | Date | Keying Instruction | Graphic / Key | Description | Page | of

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: And (A)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A	XX									FLDREF		MAILING			L	IST		FILD REFERENCE FILE
00020	A			1					R		MLGREFR								TEXT('Mailing List Field Reference')
00030	A										ACTNUM		5		Ø				COLHDG('Account' 'Number')
00040	A																		EDTCDE(Z)
00050	A										ACTTYP		1		Ø				COLHDG('Acct' 'Type')
00060	A																		TEXT('Acct Type 1=Bus 2=Gvt +
00070	A																		3=Org 4=Sch 5=Pvt 9=Oth')
00080	A										NAME		18						COLHDG('Name')
00090	A										ADDR	R	2					2	REFFLD(NAME)
00100	A																		COLHDG('Address') 3
00110	A										CITY	R	2					2	REFFLD(NAME)
00120	A																		COLHDG('City') 3
00130	A										STATE		2						COLHDG('State')
00140	A										ZIP		5		Ø				COLHDG('ZIP' 'Code')
00150	A																		EDTCDE(X)
00160	A										BATNUM		6		Ø				COLHDG('Batch' 'Number')
00170	A																		EDTCDE(Z)
00180	A										TRNTYP		1						COLHDG('Trans' 'Type')
00190	A																		TEXT('Trans Type A=Add +
00200	A																		C=Change D=Delete')

*Number of sheets per pad may vary slightly.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/060*
Printed in U.S.A.

File | Programmer | Date | Keying Instruction | Graphic / Key | Description | Page | of

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: And (A)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A										XACTNM	R							REFFLD(ACTNUM)
00020	A										XACTTP	R							REFFLD(ACTTYP)
00030	A										XNAME	R							REFFLD(NAME)
00040	A										XADDR	R							REFFLD(ADDR)
00050	A										XCITY	R							REFFLD(CITY)
00060	A										XSTATE	R							REFFLD(STATE)
00070	A										XZIP	R							REFFLD(ZIP)
00080	A										TRNNUM		5		Ø				COLHDG('Transaction' 'Number')
00090	A																		EDTCDE(Z)
00100	A										MLGLKL		3		Ø				COLHDG('Lock' 'Control')
00110	A																		TEXT('Control Number Used for +
00120	A																		record locking')
	A																		
	A																		
	A																		
	A																		
	A																		

*Number of sheets per pad may vary slightly.

Figure F-1. DDS for a Field Reference File

Example 2. Physical File with a New Record Format

The following physical file (called CUSMSTP for customer master physical file) describes the fields physically present in the data base. The following numbered items apply to Figure F-2:

1. At the file level, the REF keyword refers CPF to the physical file MLGREFP, which is a field reference file for this data base.

2. At the record level, R in position 17 specifies that CUSMST is the record format name of the record in this file. (There can only be one record format in a physical file.)

3. At the field level, Rs in position 29 specify that the attributes of fields of the same name in the REF file are to be used as attributes of these fields.

4. The field SEARCH is not defined in MLGREFP; therefore its field attributes are specified here.

5. At the key field level, K in position 17 specifies that CUST is the key field for the file.

IBM International Business Machines Corporation — DATA DESCRIPTION SPECIFICATIONS — GX21-7754-1 UM/060* Printed in U.S.A.

File | Keying Instruction: Graphic | Description | Page of

Programmer | Date | Keying Instruction: Key

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (M/O/I/B/H/M)	Location: Line	Pos	Functions
00010	A	*																	
00020	A	*	SAMPLE PHYSICAL FILE (CUSMSTP)																
00030	A	*																	
00040	A																		1 REF(MLGREFP)
00050	A					2			R		CUSMST								TEXT('Customer Master Record')
00060	A										ACTNUM	R							
00070	A										NAME	R							
00080	A										ADDR	R							
00090	A										CITY	R							3
00100	A										STATE	R							
00110	A										ZIP	R							
00120	A					4					SEARCH		10	0					
00130	A					5			K		ACTNUM								
	A																		
	A																		
	A																		

Figure F-2. DDS for a Physical File

Example 3. Physical File Using FORMAT and a Keyed Sequence Access Path

The following numbered items refer to Figure F-3:

1. The UNIQUE keyword specifies that no records with duplicate key values can be placed into the physical file CUSMSTP2.

2. The FORMAT keyword causes the physical file CUSMSTP2 to have the same record format as the physical file CUSMSTP.

3. The key fields specify a composite key for the new file CUSMSTP2.

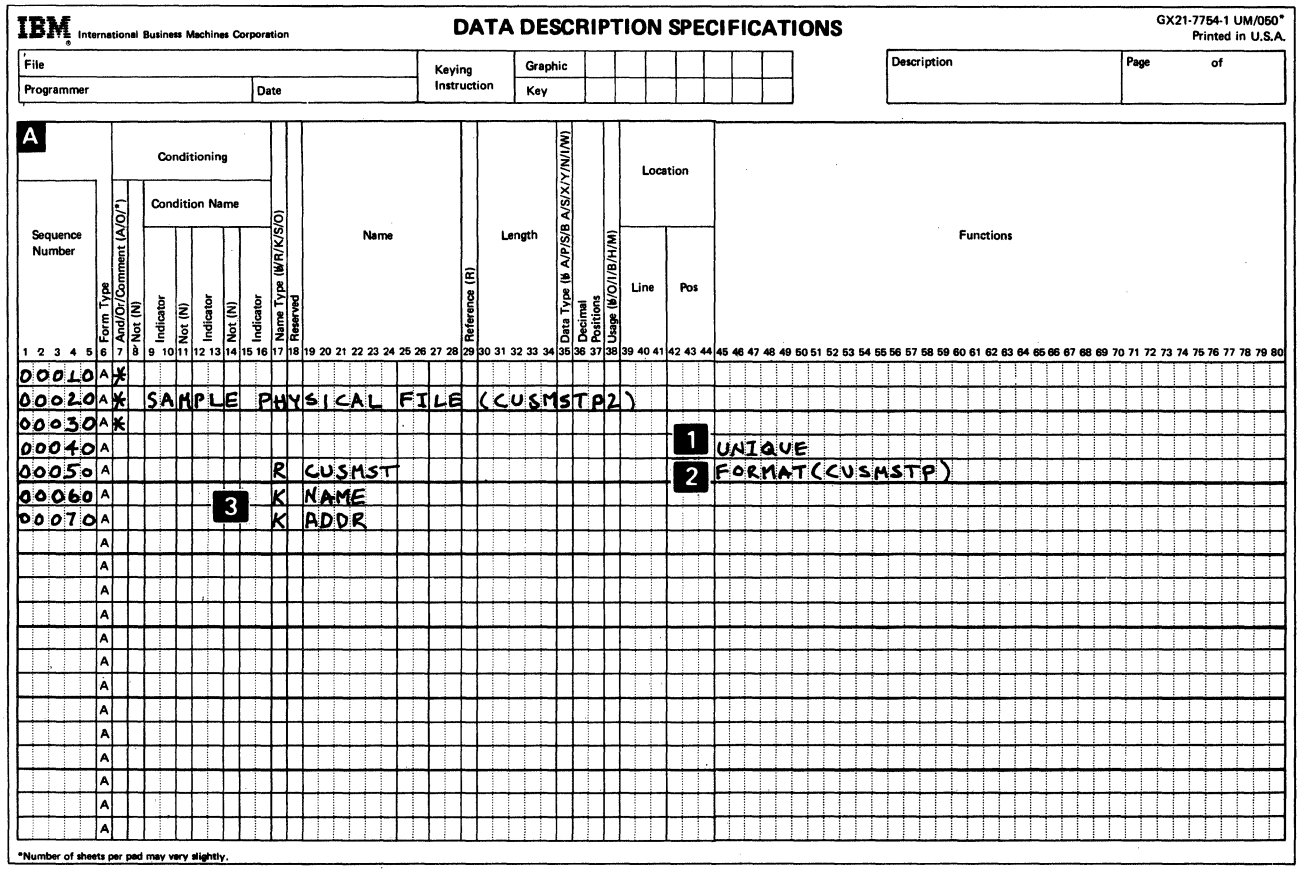


Figure F-3. DDS for a Physical File

Note: See the FORMAT keyword description in Chapter 2, *Physical Files*, for an example of a physical file with an arrival sequence access path.

Example 4. Logical File Specifying New Keys

The following numbered items refer to Figure F-4:

1. The two record formats (CUSMST1 and CUSMST2) in this logical file are based on the same physical file (CUSMSTP).

2. Record format CUSMST1 has a key different from record format CUSMST2, providing the application program with a different sequence of the same records.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050* Printed in U.S.A.

File | Keying Instruction | Graphic / Key | Description | Page of

Programmer | Date

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Indicator / Not (N) / Indicator / Not (N) / Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/I/O/B/H/M)	Location: Line	Location: Pos	Functions
00010	A	*												
00020	A	*	SAMPLE LOGICAL FILE											
00030	A	*												
00040	A			R		CUSMST1							42	1 PFILE(CUSMSTP)
00050	A					ACTNUM								
00060	A					NAME								
00070	A					STATE								
00080	A		2	K		ACTNUM								
	A													
00090	A			R		CUSMST2							42	1 PFILE(CUSMSTP)
00100	A					ACTNUM								
00110	A					NAME								
00120	A					STATE								
00130	A		2	K		NAME								

*Number of sheets per pad may vary slightly.

Figure F-4. DDS for a Logical File Specifying New Keys

Example 5. Logical File Specifying a New Record Format

The following logical file (called CUSMSTL2 for customer master logical file two) references some of the fields in the physical file CUSMSTP. Another logical file could name all fields, name fields in other physical files as well, concatenate fields, change the order of fields, rename fields, or choose different key fields. In this logical file, the programmer merely omitted some fields from the physical file. Note that this logical file is the sample used for the compiled source listing at the end of this appendix.

The following numbered items apply to Figure F-5:

1. The keyword PFILE (required for logical files) references CUSMSTP. In a physical file or in any device file, use REF (at the file level) instead of PFILE to reference other files.

2. The field names do not have R specified in position 29 as they would in physical files or in any device file.

3. As in CUSMSTP, the field ACTNUM is treated as a key field.

IBM International Business Machines Corporation — DATA DESCRIPTION SPECIFICATIONS — GX21-7754-1 UM/050* Printed in U.S.A.

File | Keying Instruction: Graphic / Key | Description | Page of

Programmer | Date

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/M)	Decimal Positions	Usage (M/O/I/B/H/N/M)	Location: Line	Pos	Functions
00010	A	*																	
00020	A	*	SAMPLE LOGICAL FILE (CUSMSTL2)																
00030	A	*																	
00040	A																		UNIQUE
00050	A								R		CUSREC							1	PFILE(CUSMSTP)
00060	A																		TEXT('Logical File Master Record')
00070	A										ACTNUM		2						
00080	A										NAME								
00090	A										ADDR								
00100	A				3				K		ACTNUM								

*Number of sheets per pad may vary slightly.

Figure F-5. DDS for a Logical File

Example 6. Simple Menu

The following menu, described in the DDS in Figure F-6 would be displayed by issuing an output operation to the record format MENU.

```
Order Department Clerk Menu

1.  General menu
90. Sign-off

Option: __
```

The following display file (named ORD005CD) contains the DDS for the order department clerk menu.

The following numbered items apply to Figure F-6:

1. The menu is displayed when a program issues an output operation to the record format MENU.

2. Five fields are displayed. The first four are constant fields that appear as words on the screen.

3. The fifth field displayed is RESP. It is an input-only field and defaults to a single underline. (If Y is not specified in position 35, an input-capable field has an additional underscore to the right, for the sign.)

4. Specifying VALUES(1 90) with field RESP causes validity checking to fail if any number other than 1 or 90 is keyed into RESP and entered.

5. Specifying BLINK on the record format causes the cursor to blink. (The cursor is positioned by default to the input-capable field RESP.)

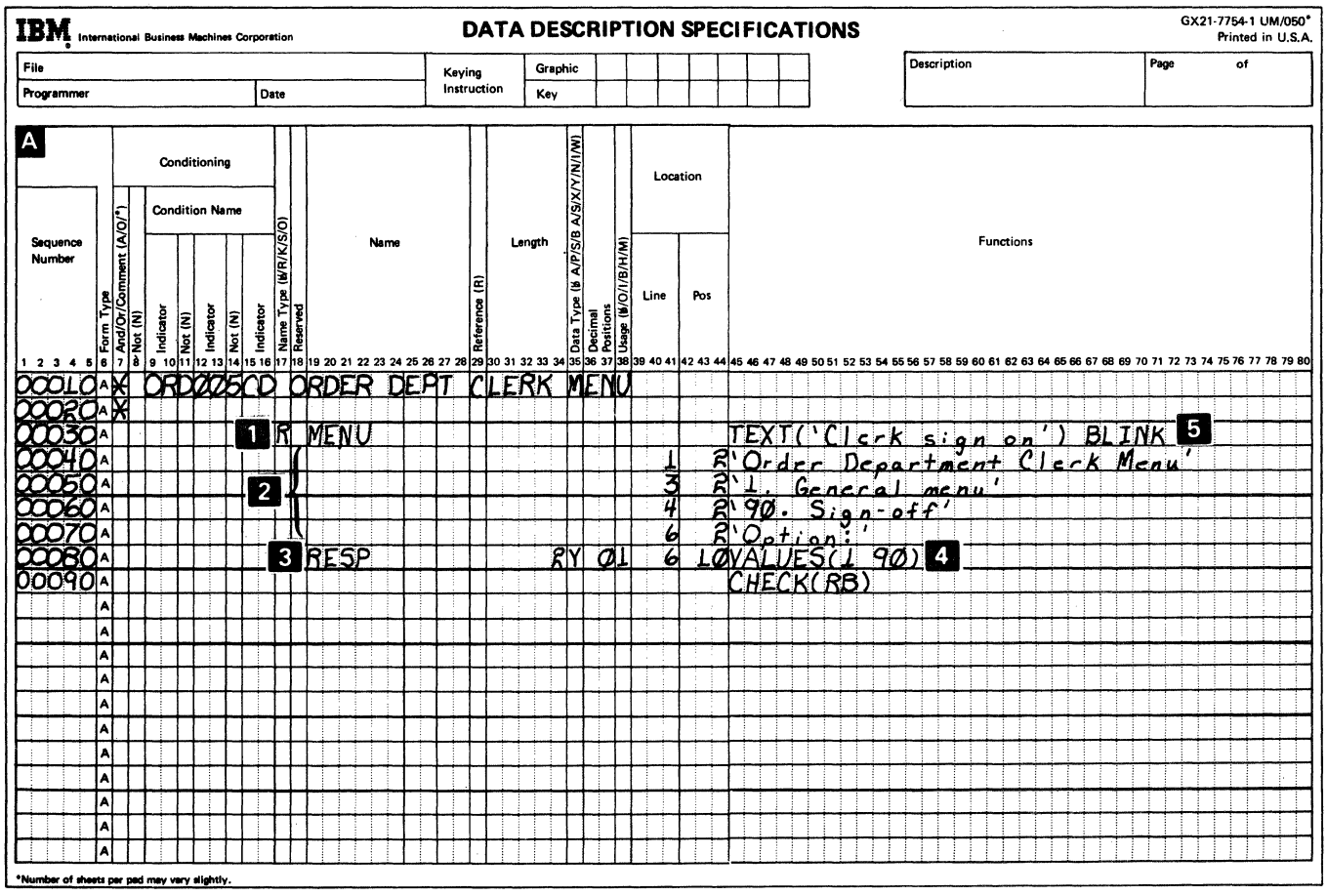
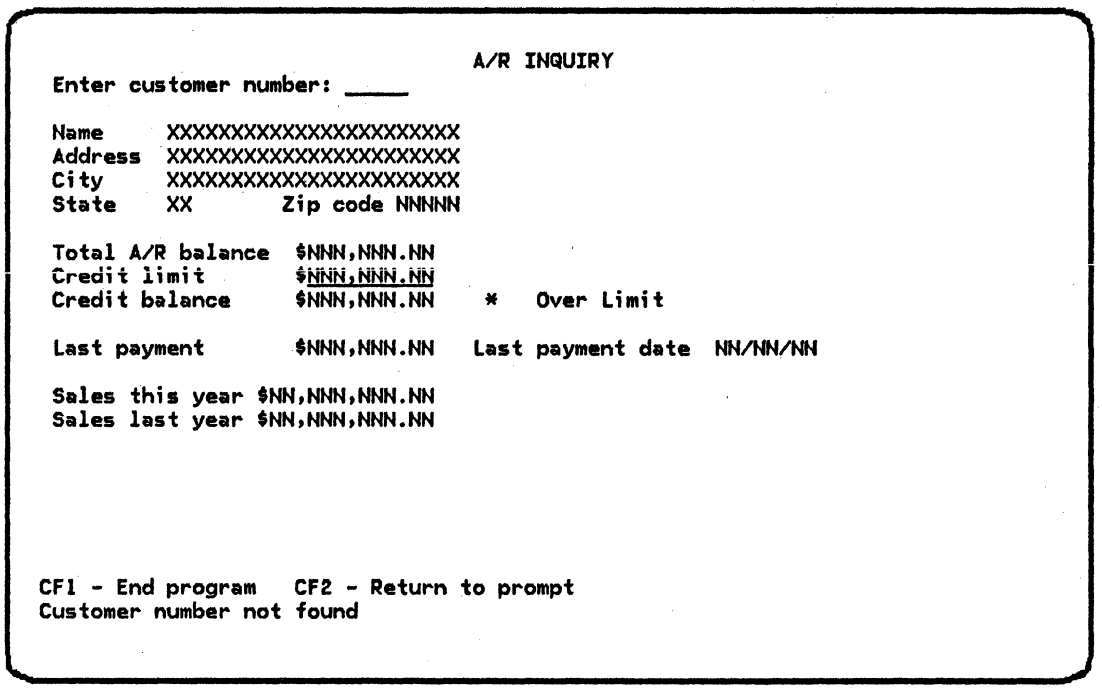


Figure F-6. DDS for a Simple Display File

Example 7. Inquiry Display with Two Record Formats

The following display is defined by the DDS in Figure F-7. It is displayed by output operations to the record formats PROMPT and RESPONSE.



The following numbered items refer to Figure F-7:

1 The PRINT keyword allows the work station user to print the display at any time by pressing the Print key.

2 An application program would display a prompt by issuing an output operation to the record format PROMPT, displaying the constant fields 'A/R INQUIRY' and 'Enter customer number:', and the named field CUST.

3 The CHECK(MF) keyword specifies that when the work station user keys into one position of the field CUST, he must key into all five positions before pressing the Enter/Rec Adv key, or an error message is displayed and the keyboard is locked. The user must press the Reset key and rekey through the input field.

4 If record format PROMPT is displayed and your program sets on indicator 40 when an output operation is issued to record format PROMPT, the error message 'Customer number not found' is displayed on the message line (line 24 of the 24-line display unless the MSGLOC keyword is specified). The message is highlighted, field CUST is displayed with its image reversed, and the keyboard locked until the work station user presses the Reset key.

5 After the work station user presses the Enter/Rec Adv key, the application program retrieves the desired information from the data base and issues an output operation to record format RESPONSE, displaying the fields described by items 8, 9, and 11.

6 The OVERLAY keyword specifies that an output operation to this record format (RESPONSE) does not cause the entire screen to be erased, as it would be by default.

7 Five constant fields ('Name', 'Address', 'City', 'State', and 'Zip Code') and five named fields (NAME, ADDRESS, CITY, STATE, ZIP) are grouped together on the display by the line/position specifications. NAME, ADDRESS, CITY, and STATE default to character-type fields (A in position 35) because no decimal positions are specified. ZIP is a numeric-only, integer field (Y in position 35; 0 in position 37), so its display length equals its specified length.

8 Seven named fields have constant fields located near them on the display to identify them as in 7. In addition, edit codes EDTCDE(2) and EDTCDE(Y) affect this display. EDTCDE(2) is used for monetary amounts. EDTCDE(Y) is used for dates.

9 The DSPATR(UL) keyword underlines the output field CRDLMT.

10 If indicator 30 is set on when an output operation is issued to record format RESPONSE, then the constant field '*Over Limit' is displayed highlighted and blinking.

11 Instructions to the work station user are generally located at the bottom of the display, just above the message line.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050* Printed in U.S.A.

File		Keying Instruction	Graphic								Description	Page of
Programmer	Date		Key									

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name or (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (M/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A																1	42	PRINT
00020	A																		CA01(21 'End program')
00030	A																		CA02(22 'Display PROMPT')
00040	A				2				R		PROMPT								
00050	A																1	35	'A/R INQUIRY'
00060	A														4		2	2	'Enter customer number:'
00070	A	3									CUST		5	Y	0	B		+1	CHECK(MF)
00080	A		40																ERRMSG('Customer number not +
00090	A														5				found' 40)
00100	A	*																	
00110	A				6				R		RESPONSE								OVERLAY 7
00120	A																4	2	'NAME'
00130	A										NAME		23				4	10	
00140	A																5	2	'Address'
00150	A										ADDRES		23				5	10	
00160	A																6	2	'City'
00170	A										CITY		23				6	10	
00180	A																7	2	'State'
00190	A										STATE		2				7	10	
00200	A																7	19	'Zip code'
00210	A										ZIP		5	Y	0			+1	
00220	A	*																	
00230	A																9	2	'Total A/R balance'
00240	A										ARBAL		8		2		9	21	EDTCDE(2 $)
00250	A																10	2	'Credit limit'
00260	A										CRDLMT		8		2		10	21	EDTCDE(2 $) DSPATR(UL) 10
00270	A																11	2	'Credit balance'
00280	A										CRDSAL		8		2		11	21	EDTCDE(2 $)
00290	A		30	11														+4	'* Over Limit' DSPATR(HI BL)
00300	A	*																	
00310	A																13	2	'Last payment'
00320	A										LSTPMT		8		2		13	21	EDTCDE(2 $)
00330	A																	+3	'Last payment date'
00340	A										LPDAT		6		0			+2	EDTCDE(Y)
00350	A																15	2	'Sales this year'
00360	A										THSYR		10		2			+1	EDTCDE(2 $)
00370	A																16	2	'Sales last year'
00380	A										LSTYR		10		2			+1	EDTCDE(2 $)
00390	A																23	2	'CF1 - End program CF2 - Return +
00400	A																		to prompt'
	A																		
	A																		

*Number of sheets per pad may vary slightly.

Figure F-7. Display with Two Record Formats

Example 8. Work Station Display Showing Subfile Operations

The following displays illustrate what a work station user would see when using a subfile. The user answers an initial prompt, and the program displays the contents of a subfile.

1. The initial prompt is displayed when the program issues an output operation to the record formats PROMPT and KEYS (lines 00800 and 03700 of Figure F-8 [the DDS for this display]):

The initial prompt PROMPT (line 00500)

```
CUSTOMER REVIEW PROMPT

Customer name to start review  _____________________
```

Record format KEYS

```
CF1 - End program
```

2. The user keys the name BAKER in the input field and presses the Enter/Rec Adv key:

```
CUSTOMER REVIEW PROMPT

Customer name to start review  BAKER_____________

CF1 - End program
```

3. The program fills the subfile by reading records from a data base file and writing them to the subfile. When the subfile is full, the program issues an output operation to the subfile control record format SFLCTLR (see Figure F-8).

   The name Baker does not actually exist on the customer review; the program displays the next name in the alphabetical listing followed by as much of the customer review list as fits in one subfile page.

The subfile control record SFLCTLR (line 02500)

The subfile record SFLR (line 02100)

Record format KEYS (line 03700)

```
CUSTOMER REVIEW

Name                      Address                 Number

BOOTRILL, J. HOWARD       975 E. JOHNSON          10015
BOSWORTH, NORMAN          4540 BOTH PLACE         20111
BREYER, SUSAN             124 19TH STREET         10013
COLLIER, LAWRENCE B.      777 BROADWAY            10014
CONSTANTS, JAMES A.       896 S. MAIN             20114
COOKSEY, JAMES            674 W. 14TH ST          20117
GERSALLA, BEVERLY         832 NE 12TH ST          20115
HARRINGWORTH, JOHN        1515 W. NICOLLET        20116
HARVELD, DAVID G.         RR1                     10016
HEIMER, JUDITH A.         4110 S. ELM             10019
JOHN, J. YANG             5123 W 56TH ST          10011
KAMP, HENRY               376 W. ALICE            20112
KESLEY, ROBERT S.         356 N. PACIFIC          10008
KYLE, JENNIFER A.         722 N. RIVER DRIVE      10009
LOWREY, THOMAS A.         135 19TH PLACE          10004
MC MILLER, SUSAN          367 FOREST              10014
NOVACS, JEFFREY A.        8080 NW 3RD PLACE       10010

CF1 - End program
```

4. Because all names in the customer review do not fit on a single display, the user presses the CF3 key to display more data. The program then refills the subfile with more records, redisplays the subfile, sets option indicator 50 on, and redisplays KEYS:

```
CUSTOMER REVIEW

Name                      Address                 Number

ROTHBERG, MELVIN          321 Circle Drive        20113
SMITH, RICHARD            111 15TH AVE            10003
STONE, HIRAM A.           345 LAWRENCE            1005
THOMPSON, JESSE W.        5005 W 50TH             1006
WILSON, MARGARET A.       3131 E. 40TH AVE        1007









CF1 - End program   CF3 - Roll up   CF4 - First Display
```

The following display file (Figure F-8) contains the DDS for the preceding displays. The first coding sheet shows the prompt and two messages; the second coding sheet shows the subfile record format and subfile control record format.

The following numbered items apply to part 1 of Figure F-8:

1. Specifying CA01 (on line 00500) enables the work station user to return control to the program by pressing the CF1 key. Pressing the CF1 key sets response indicator 91 on. The CAnn keyword is specified instead of CFnn because CAnn does not pass input data to the program. If it were necessary to pass input data to the program, CFnn would be specified.

2. When the program issues an output operation to the record format PROMPT, four fields are always displayed: two constant fields (on lines 01000 and 01200) and one output/input field (NAME on line 01400).

3. If the program issues an output operation to the record format PROMPT with the indicator 61 on, the error message specified with the ERRMSG keyword is displayed along with the four other fields in the record format.

   The keyboard is locked, the NAME field is displayed with its image reversed, and the error message appears on the message line (line 24). The work station user must press the Reset key and rekey the NAME field to escape the error condition.

4. As in 3, if the program issues an output operation to the record format PROMPT with indicator 62 set on, the constant field specified on lines 02000 through 02200 is displayed as a highlighted field.

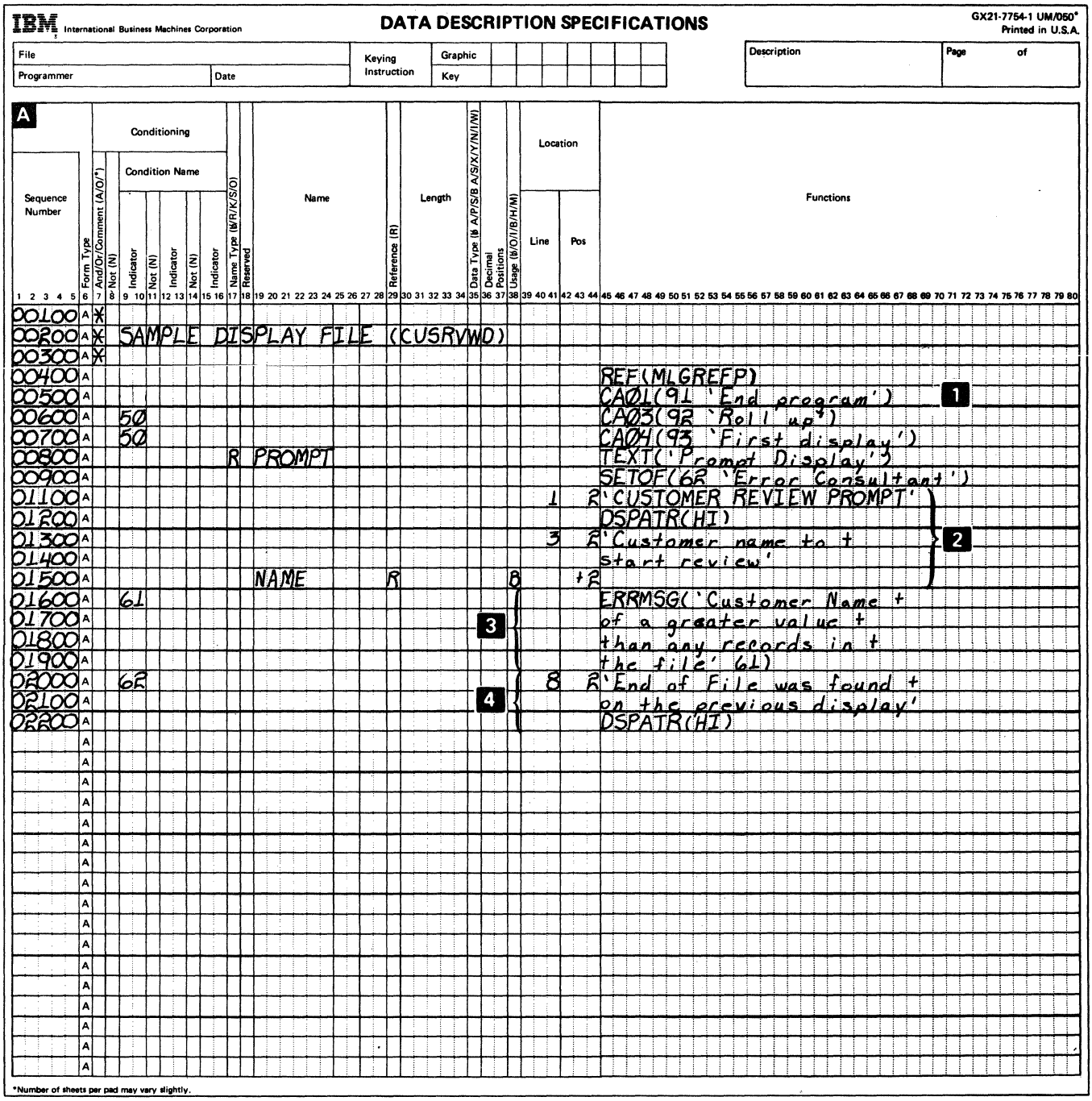


Figure F-8 (Part 1 of 2). Customer Review

The following numbered items apply to part 2 of Figure F-8:

5 The subfile record format describes each record in the subfile. Each record has three fields, NAME, ADDR, and ACTNUM, which are referenced to the field reference file MLGREFP (see REF keyword in part 1 of Figure F-8).

6 The first record of the subfile is displayed on line 5, with its fields NAME, ADDR, and ACTNUM beginning in positions 2, 27, and 52, respectively. Succeeding records of the subfile appear on the lines below it (see the display for step 3).

7 The subfile control record format name is SFLCTLR. The fields associated with it (in this case, the four constant fields on lines 03200 through 03600) are displayed when the program issues an output operation to it.

8 When the program sets option indicator 60 off and issues an output operation to the subfile control record format SFLCTLR, the subfile is cleared (SFLCLR). When the program sets option indicator 60 on and issues an output operation to the subfile control record format SFLCTLR, the subfile is displayed (SFLDSP) and the subfile control record is displayed (SFLDSPCTL).

9 Because SFLSIZ and SFLPAG are equal, your program must handle rolling the data by refilling the subfile.

10 The record format KEYS contains constant fields that are displayed at the bottom of the display (one line above the message line) to give instructions to the work station user. These constant fields are not specified in the subfile control record format because fields in the subfile control record format would then be both above and below fields in the subfile record format, and that is invalid.

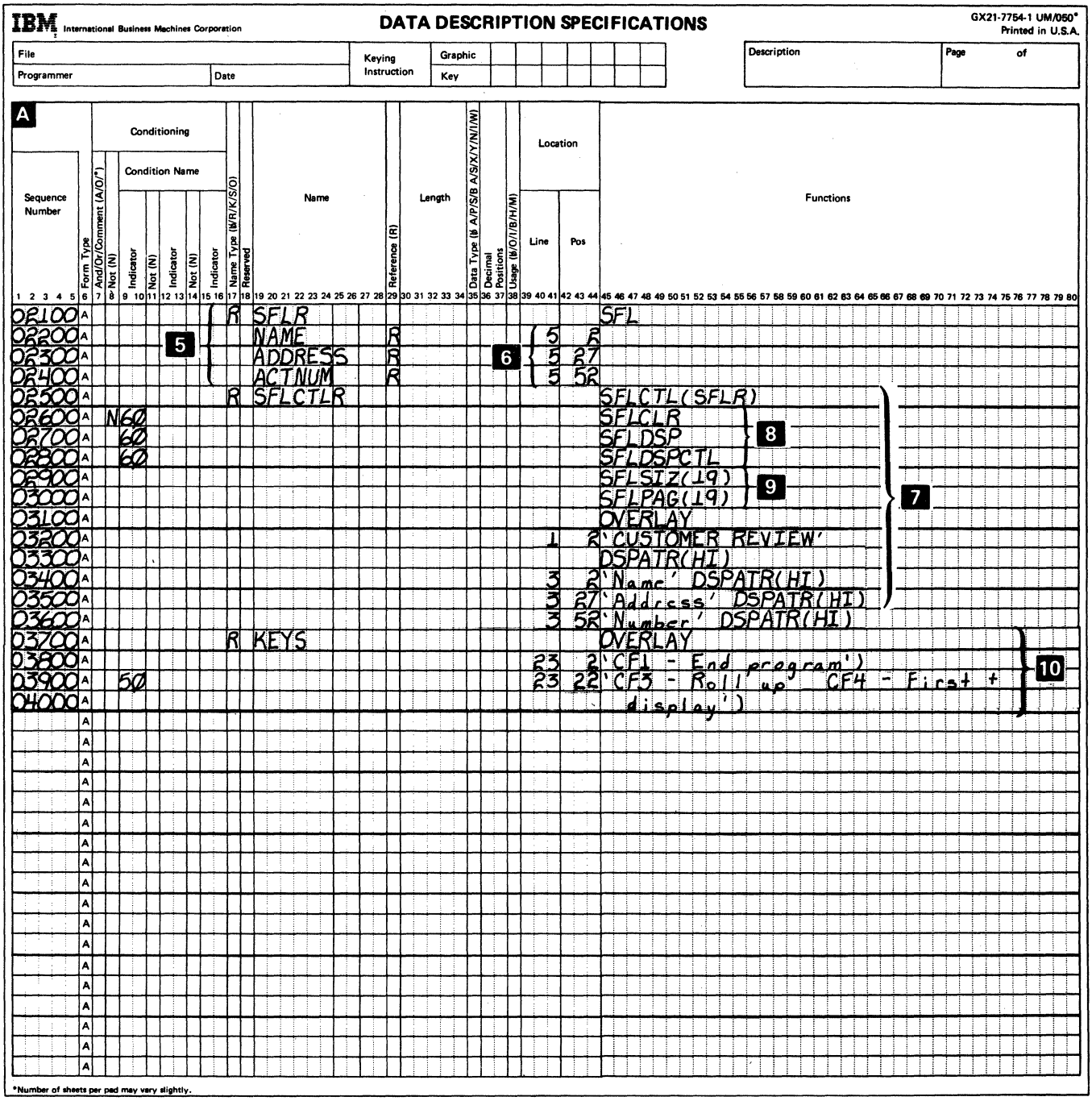
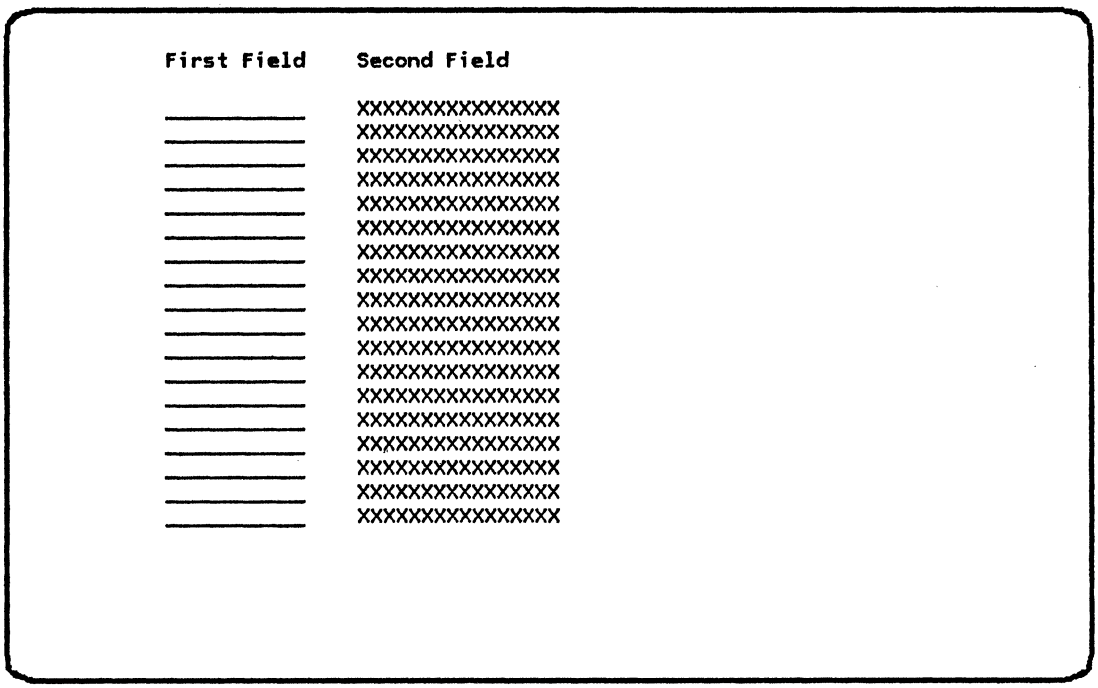


Figure F-8 (Part 2 of 2). Customer Review

Example 9. Subfile with SFLPAG Value Equal to SFLSIZ Value

The following display is defined by the DDS in Figure F-9. It is displayed by an output operation to the subfile control record format SFLCTL1.



The following numbered items refer to Figure F-9:

1. The subfile record format SFL1 and the subfile control record format SFLCTL1 together define one subfile. The parameter value for the SFLCTL keyword is the name of the subfile record format.

2. Each subfile record is made up of two fields: FLD1 and FLD2. FLD1 is 10 bytes long (11 bytes display length because it defaults to signed numeric); FLD2 is 16 bytes long. FLD1 is an input-only field; FLD2 is an output-only field. Eighteen subfile records would appear on the screen, with the first one on line 3 and the last one on line 20. For each subfile record on the screen, two fields (FLD1 and FLD2) would appear, with four spaces between FLD1 and FLD2.

3. SFLSIZ and SFLPAG (required keywords) have equal values (18). Therefore one page equals the whole subfile. For all subfiles, the value of the SFLPAG keyword is the number of subfile records displayed at any one time (unless the SFLDROP keyword or variable length records are used).

4. SFLDSP (a required keyword) and SFLDSPCTL (an optional keyword) are specified with indicator 05. Therefore, when indicator 05 is set on, the subfile and subfile control records can be displayed by an output operation to the subfile control record format SFLCTL1.

5. SFLCLR (an optional keyword) is specified with option indicator 05 preceded by an N. When indicator 05 is set off, the subfile can be cleared by an output operation to SFLCTL1.

6. ROLLUP (an optional keyword) is specified with response indicator 01, and ROLLDOWN (an optional keyword) is specified with response indicator 02. Note also that the entire subfile equals one page, which means that the whole subfile is displayed at one time. Therefore, when the work station user presses the Roll Up key, control passes to the program with indicator 01 on, and when the work station user presses the Roll Down key, control passes to the program with indicator 02 on. The program must handle rolling, by reading, clearing, rewriting, and redisplaying the subfile. Without ROLLUP and ROLLDOWN specified, the work station user would receive an error message when pressing the Roll Up or Roll Down key.

7. Two constants ('First Field' and 'Second Field') are displayed when the subfile control record is displayed (SFLDSPCTL in effect). As specified in this subfile, they act as column headings to the subfile records.

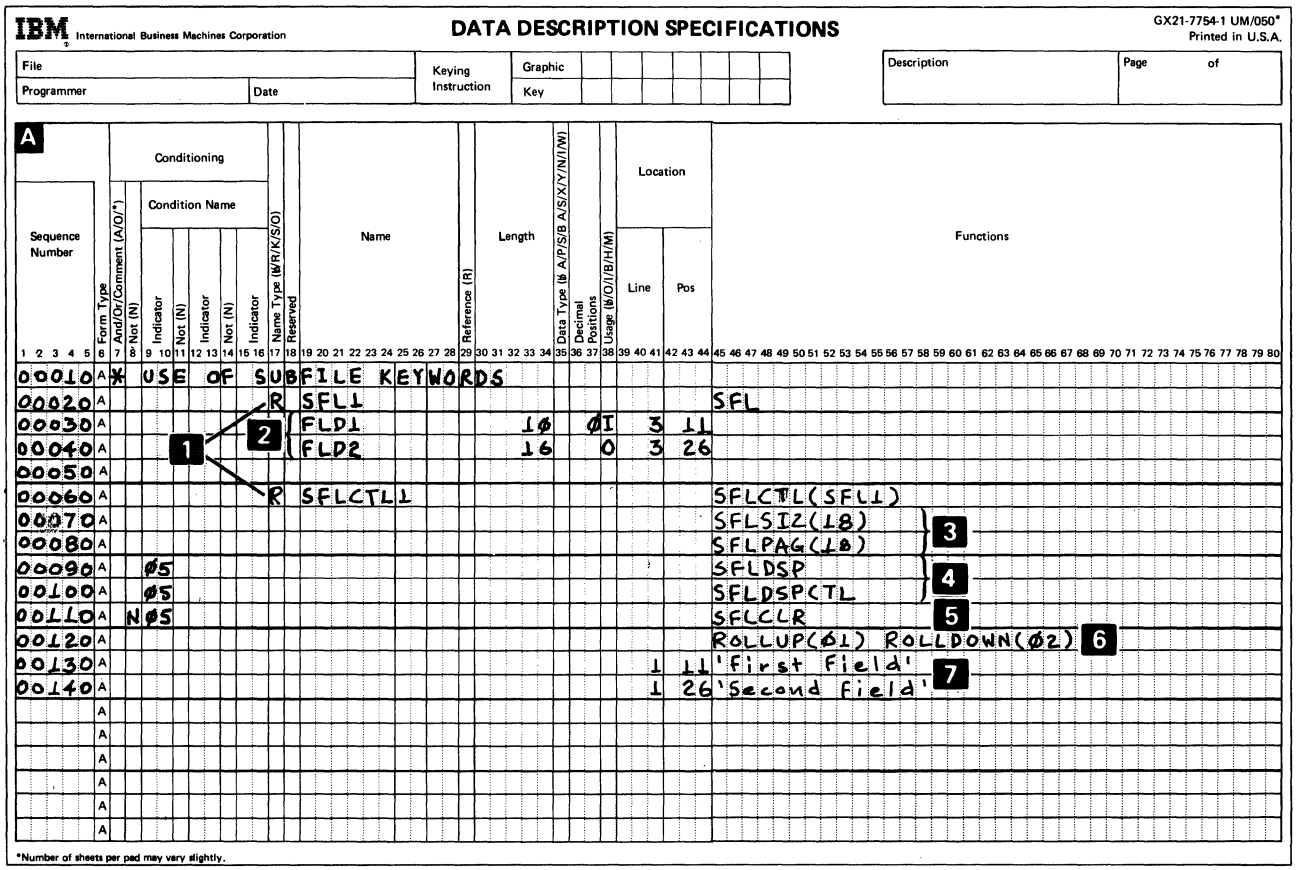


Figure F-9. Subfile with Subfile Size Equal to Subfile Page

Example 10. Subfile for Update without Validity Checking

The following display shows the subfile defined in Figure F-10. The subfile contains records to be updated. A work station user could key into as many fields as required, probably not changing all the records. When the work station user presses Enter/Rec Adv key, the program reads only the records that have been changed (issuing Get-Next-Changed operations), accepting records without validity checking and building a transaction file with the changed records. Only when the transaction file is run against a master file is validity checking performed.

```
                          INVENTORY UPDATE

        ITM NBR   DESCRIPTION        AMOUNT   UNITS

          NL01    NAILS                 6     BOX
          NL04    NAILS                18     BOX
          NL10    NAILS                 2     BOX
          NL12    NAILS                14     BOX
          NL14    NAILS                21     BOX
          NL20    NAILS                 1     BOX
          SC02    SCREWS               13     BOX
          SC04    SCREWS               10     BOX
          SC05    SCREWS               20     BOX
          SC06    SCREWS               20     BOX
          SC07    SCREWS               24     BOX
          SC08    SCREWS               20     BOX
          SC20    SCREWS               10     BOX
          SC21    SCREWS                2     BOX
          SC23    SCREWS                4     BOX
          SC25    SCREWS                4     BOX
          SC27    SCREWS                2     BOX

CF1 - End program   CF2 - Cancel updates
```

The following numbered items refer to Figure F-10:

1. CHECK(RB) is specified for the fields ITMNBR and AMOUNT so that when the work station user keys into these fields, the keyed values are right-adjusted and blank-filled in the displayed fields. CHECK(LC) is specified for the field DESCR so that the work station user can key in lowercase characters to be passed to the program. Without CHECK(LC), lowercase characters keyed into DESCR are automatically converted to uppercase characters before being passed to the program.

2. The UNLOCK keyword, specified on the subfile control record format, with the parameter value (*ERASE *MDTOFF), permits the work station user to continue keying in immediately after pressing the Enter/Rec Adv key. The program does not have to issue another output operation to the subfile control record format after issuing an input operation to unlock the keyboard.

3. A separate record format KEYS is necessary because the field locations for any one record format cannot interrupt the field locations for any other record format.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050* Printed in U.S.A.

File		Keying Instruction	Graphic								Description	Page	of
Programmer	Date		Key										

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (B/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (B A/P/S/B A/S/X/Y/N/I/N)	Decimal Positions	Usage (B/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A	*																	
00020	A	*									SUBFILE FOR UPDATE WITHOUT VALIDITY CHECKING								
00030	A	*																	
00040	A																		CF01(10 'End program')
00050	A																		CF02(11 'Clear and redisplay + subfile')
00060	A																		
00070	A								R		SFLR								SFL
00080	A										ITMNBR		7			B	5	12	CHECK(RB)
00090	A										DESCR		15					+3	CHECK(LC)
00100	A										AMOUNT		6Y	02		B		+3	CHECK(RB)
00110	A										UNITS		5			B		+3	
00120	A	*																	
00130	A								R		SFLCTLR								SFLCTL(SFLR) OVERLAY
00140	A																		SFLSIZ(17)
00150	A																		SFLPAG(17)
00160	A			01															SFLDSP
00170	A			02															SFLDSPCTL
00180	A			03															SFLCLR
00190	A																		UNLOCK(*ERASE *MDTOFF)
00200	A	*																	
00210	A																1	33	'INVENTORY UPDATE'
00220	A																3	12	'ITM NBR'
00230	A																3	22	'DESCRIPTION'
00240	A																3	40	'AMOUNT'
00250	A																3	44	'UNITS'
00260	A	*																	
00270	A								R		KEYS								OVERLAY
00280	A					3											23	2	'CF1 - End program CF2 - Cancel + updates'
00290	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

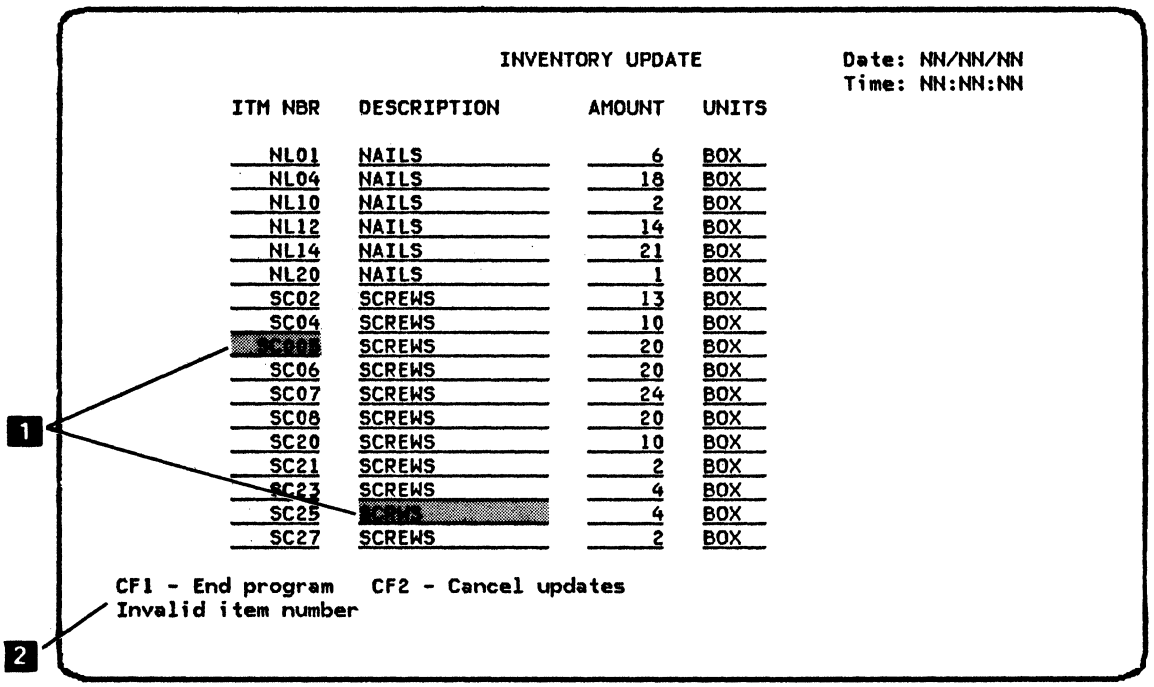
1 2 3

*Number of sheets per pad may very slightly.

Figure F-10. Subfile for Update without Validity Checking

Example 11. Subfile for Update with Validity Checking

The following display shows the subfile defined in Figure F-11. The subfile contains records that the work station user has tried to update. Two of them 1 have failed a validity check performed by the program. The two fields in error are displayed with their images reversed, and the first relevant error message 2 is displayed on the message line (line 24).



The following numbered items refer to Figure F-11:

1. When the work station user presses the Enter/Rec Adv key, the program issues Get-Next-Changed operations to the subfile record format SFLR. If the data keyed into the fields fails any validity checking performed by the program, the program can issue update operations to the records that have failed, setting on indicator 15 each time and setting indicators 16, 17, 18, and 19 as necessary to select the DSPATR(RI PC) keywords and the SFLMSG keywords for the fields in error.

   When the program displays the subfile by issuing an output operation to SFLCTLR, the fields selected have their images reversed, the cursor is positioned to the first selected field on the display, and the error message applying to the first field selected is displayed on the message line (line 24).

2. Four constant fields are specified for date and time. They are 'Date:' and 'Time:' and the two fields associated with the DATE and TIME keywords (identified only by their locations).

3. A separate record format KEYS is specified so that the instructions describing valid command keys can be displayed at the bottom of the display. (The constant fields displaying these instructions cannot be specified on the subfile control record format if the locations of the fields are below the subfile records.) Note that the CFnn keywords are specified at the file level, so they are always in effect.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050* Printed in U.S.A.

File		Keying Instruction	Graphic								Description	Page	of
Programmer	Date		Key										

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (H/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/1/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A	*		SUBFILE		FOR		UPDATE			WITH VALIDITY CHECKING								
00020	A	*																	
00030	A																		CF01(21 'End program')
00040	A																		CF02(22 'Clear and redisplay +
00050	A																		subfile')
00060	A								R		SFLR								SFL
00070	A			15															SFLNXTCHG
00080	A										ITMNBR		7	Y	0B		5	12	CHECK(RB) COMP(GT 0)
00090	A			16															DSPATR(RI PC)
00100	A										DESCR		15		B			+3	CHECK(LC)
00110	A			17															DSPATR(RI PC)
00120	A										AMOUNT		6	Y	0B			+3	CHECK(RB) COMP(GT 0)
00130	A			18															DSPATR(RI PC)
00140	A										UNITS		5		B			+3	COMP(NE ' ')
00150	A			19															DSPATR(RI PC)
00160	A	*																	
00170	A								R		SFLCTLR								SFLCTL(SFLR) OVERLAY
00180	A																		SFLSIZ(17)
00190	A																		SFLPAG(17)
00200	A			01															SFLDSP
00210	A			02															SFLDSPCTL
00220	A			03															SFLCLR SFLINZ
00230	A			16															SFLMSG('Invalid item number')
00240	A			17													1		SFLMSG('Invalid description')
00250	A			18															SFLMSG('Invalid amount')
00260	A			19															SFLMSG('Invalid units')
00270	A																1	33	'INVENTORY UPDATE'
00280	A																3	12	'ITM NBR'
00290	A																3	22	'DESCRIPTION'
00300	A																3	40	'AMOUNT'
00310	A																3	49	'UNITS'
00320	A																1	60	'Date:'
00330	A														2		1	66	DATE EDTCDE(Y)
00340	A																2	60	'Time:'
00350	A																2	66	TIME
00360	A	*																	
00370	A								R		KEYS								OVERLAY
00380	A				3												R3	2	'CF1 - End program CF2 - Cancel +
00390	A																		updates'

*Number of sheets per pad may vary slightly.

Figure F-11. Subfile for Update with Validity Checking

Example 12. Subfile with SFLINZ and SFLRNA Keywords

The following display shows a subfile defined in Figure F-12. The subfile has been initialized using the SFLINZ and SFLRNA keywords. The subfile is displayed, and either the program or the work station user can enter records. (At this point the records are inactive, although the subfile is active.)

```
                SUBFILE WITH SFLINZ AND SFLRNA KEYWORDS

FIELD ONE     FIELD TWO     FIELD THREE

_____         00000         Value
_____         00000         Value
_____         00000         Value
_____         00000         Value
_____         00000         Value
_____         00000         Value




CF1 - End program
```

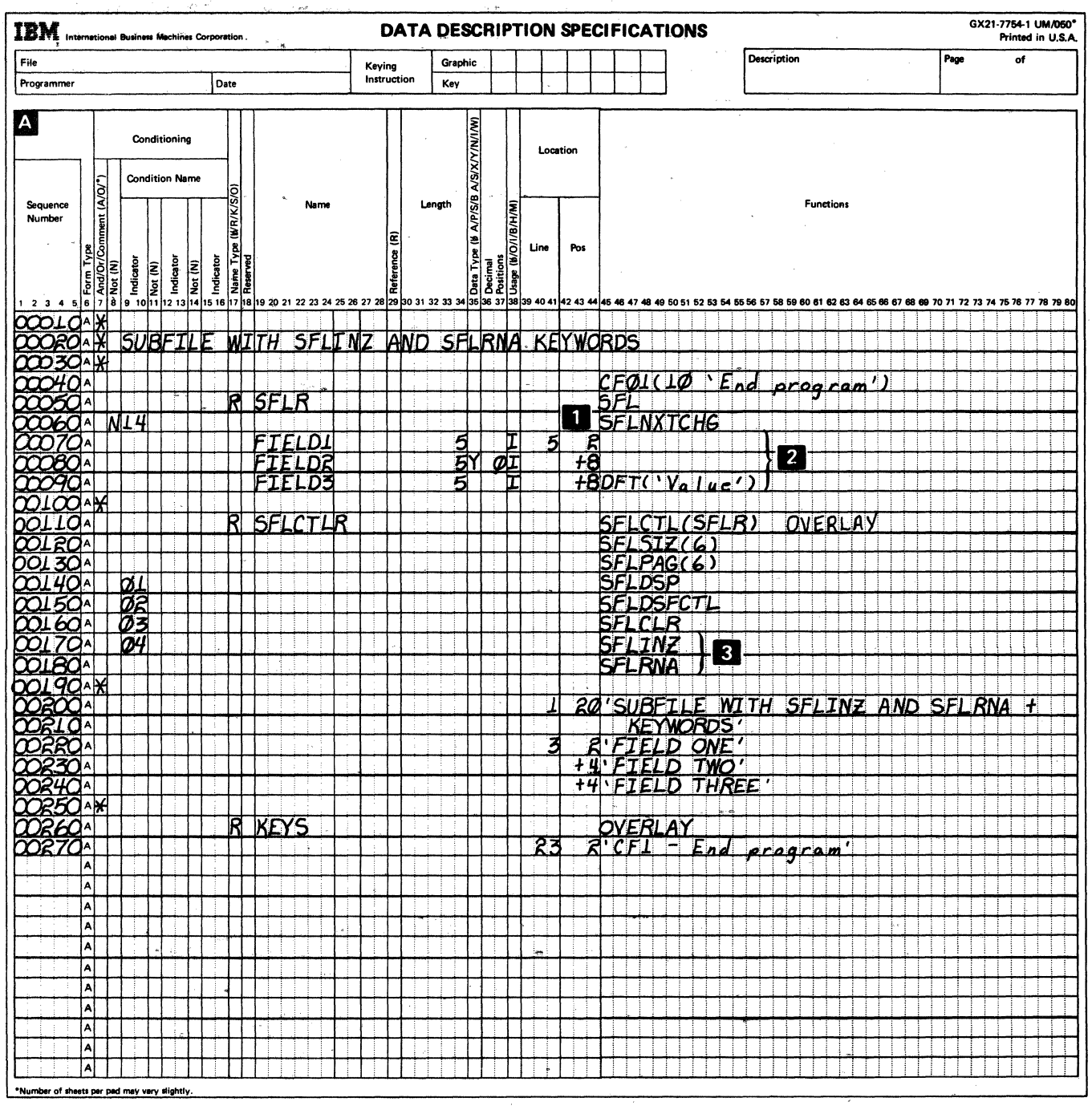
The following numbered items refer to Figure F-12:

1. For this display, the program has issued an output operation to SFLCTLR with indicators 01, 02, and 04 set on. The subfile is active; the subfile records are not active and not modified. To modify subfile records, the program can issue output operations to individual subfile records or the work station user can key changes. A Get-Next-Changed operation does not retrieve subfile records until at least one record has been modified.

2. As initialized, the three input-only fields in the subfile record format appear as follows:

Field	Type	Displayed Value
FIELD1	Character	Blanks
FIELD2	Numeric	Zeros
FIELD3	Character (constant specified)	Specified constant (*Value*)

3. The SFLINZ keyword is required for the SFLRNA keyword.



IBM International Business Machines Corporation.

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050* Printed in U.S.A.

File | Keying Instruction | Graphic / Key | Description | Page of

Programmer | Date

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blR/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blA/P/S/B A/S/X/Y/N/1/W)	Decimal Positions	Usage (blO/I/B/H/M)	Location: Line	Pos	Functions
1 2 3 4 5	6	7	8	9 10 11	12 13	14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A	X																	
00020	A	X	SUBFILE WITH SFLINZ AND SFLRNA KEYWORDS																
00030	A	X																	
00040	A																		CF01(10 'End program')
00050	A								R		SFLR								SFL
00060	A		N14															1	SFLNXTCHG
00070	A										FIELD1		5		I		5		2
00080	A										FIELD2		5Y	0	I			+8	2
00090	A										FIELD3		5		I			+8	DFT('Value') 2
00100	A	X																	
00110	A								R		SFLCTLR								SFLCTL(SFLR) OVERLAY
00120	A																		SFLSIZ(6)
00130	A																		SFLPAG(6)
00140	A			01															SFLDSP
00150	A			02															SFLDSFCTL
00160	A			03															SFLCLR
00170	A			04															SFLINZ 3
00180	A																		SFLRNA 3
00190	A	X																	
00200	A																1	20	'SUBFILE WITH SFLINZ AND SFLRNA +
00210	A																		KEYWORDS'
00220	A																3	2	'FIELD ONE'
00230	A																	+4	'FIELD TWO'
00240	A																	+4	'FIELD THREE'
00250	A	X																	
00260	A								R		KEYS								OVERLAY
00270	A																23	2	'CF1 - End program'
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

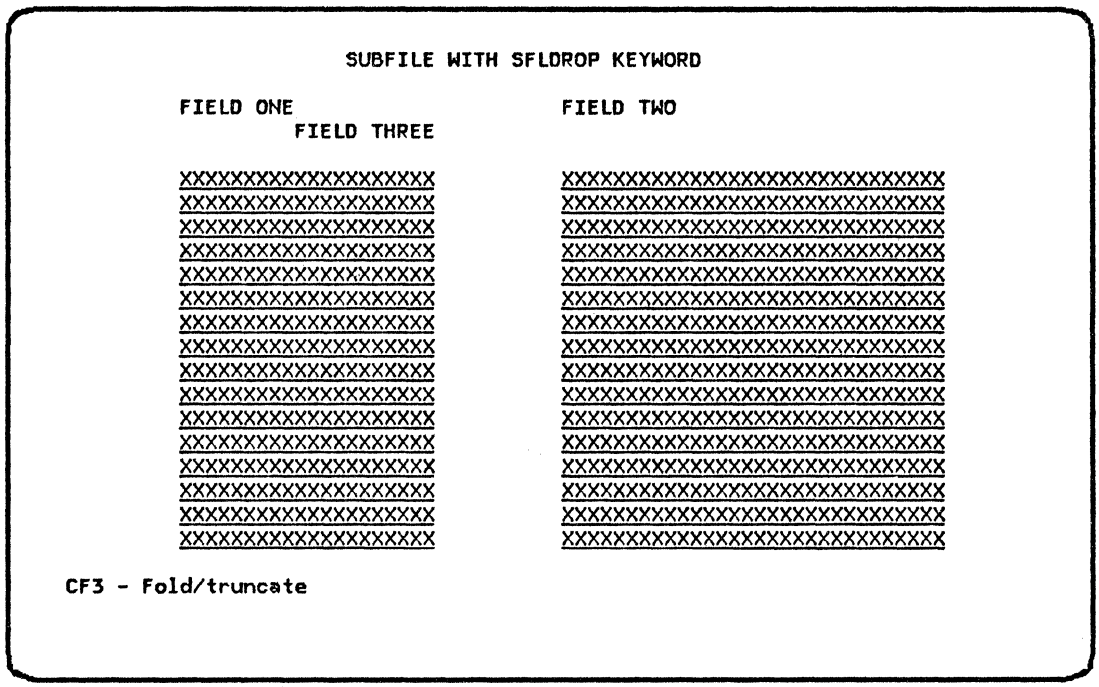
*Number of sheets per pad may vary slightly.

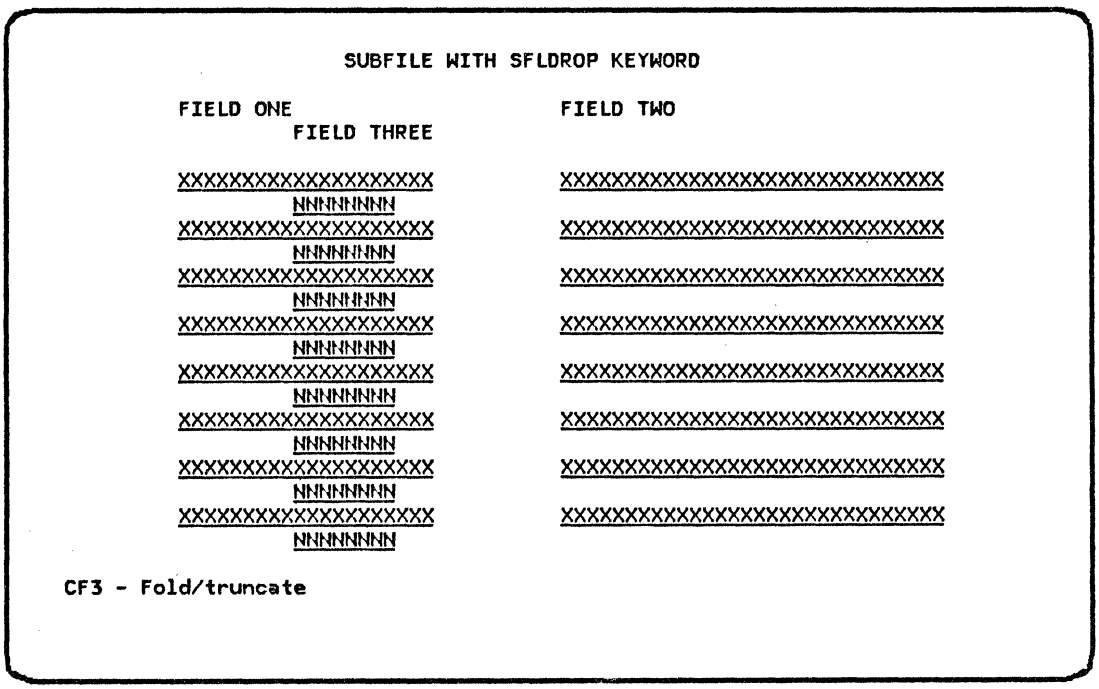
Figure F-12. Subfile with SFLINZ and SFLRNA Keywords

Example 13. Subfile with SFLDROP Keyword

The following two displays show a subfile defined by the DDS in Figure F-13. In the first display, the records are displayed in the truncated form, as shown in this display (FIELD3 is not displayed).

The second display shows the subfile with records in the folded form (after the CF3 key has been pressed once). To switch from one form to the other, the work station user presses the CF3 key (as specified on the SFLDROP keyword).





The following numbered items refer to Figure F-13:

1 FIELD3 is specified on the subfile record format with locations on the display line following the initial line specified for subfile records; therefore, its display can be controlled by pressing CF3 (see item 2).

2 The SFLDROP keyword specifies that the CF3 key is to be the Drop key.

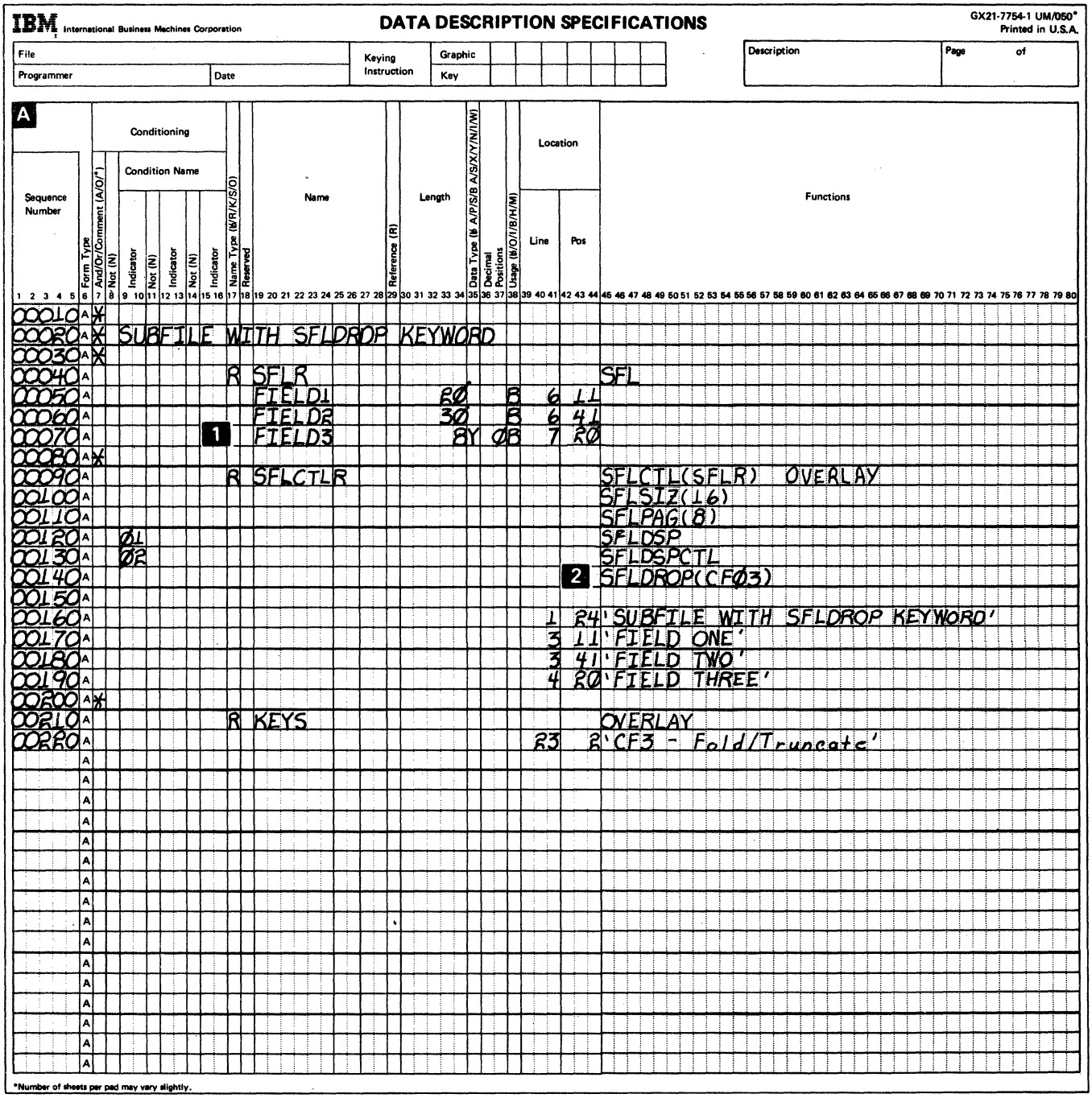


Figure F-13. Subfile with SFLDROP Keyword

Example 14. Subfile with Variable Length Records (Field Selection)

The following display shows a subfile defined by the DDS in Figure F-14. This example illustrates the use of setting option indicators (specified in the subfile record format) to vary the length of subfile records.

```
                    SUBFILE WITH VARIABLE LENGTH RECORDS

FIELD ONE                          FIELD TWO
          FIELD THREE

XXXXXXXXXXXXXXXXXXX                XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          NNNNNNNN
XXXXXXXXXXXXXXXXXXX                XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXX                XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          NNNNNNNN
XXXXXXXXXXXXXXXXXXX                XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          NNNNNNNN
XXXXXXXXXXXXXXXXXXX                XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXX                XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXX                XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          NNNNNNNN
XXXXXXXXXXXXXXXXXXX                XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          NNNNNNNN
XXXXXXXXXXXXXXXXXXX                XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXX                XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXX                XXXXXXXXXXXXXXXXXXXXXXXXXXXXX

CF1 - Return
```

The following numbered items refer to Figure F-14:

1. Option indicator 26 is specified for FIELD3. By setting option indicator 26, the program can select or not select FIELD3 when issuing output operations to SFLR (writing records to the subfile). The lengths of records in the subfile varies depending on how FIELD3 is selected.

2. You cannot control the display of the constant field 'FIELD THREE' when the work station user presses the Drop key.

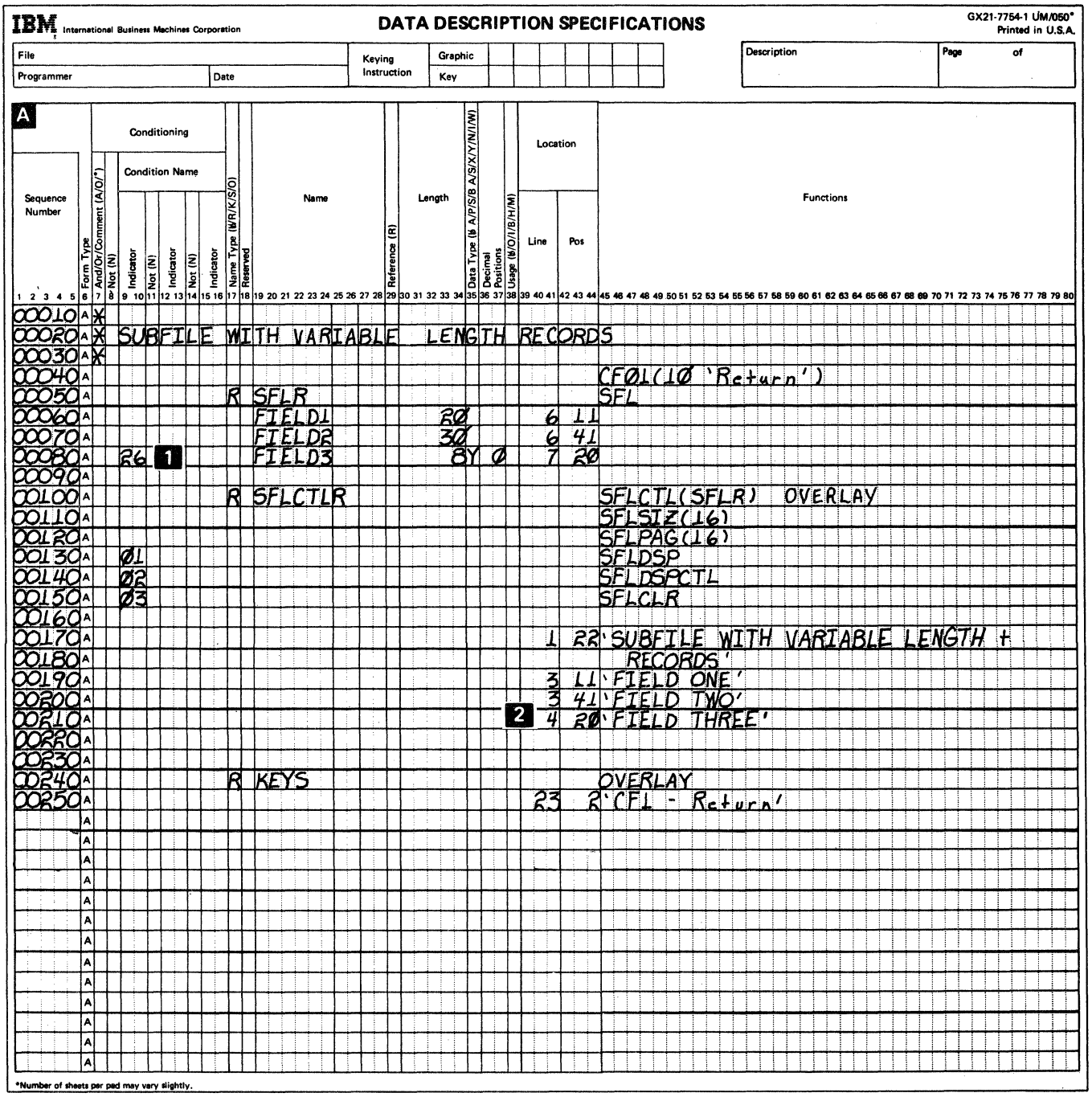


Figure F-14. Subfile with Variable Length Records (Field Selection)

Example 15. Subfile with Rolling by CPF

The following display shows a subfile that because the SFLSIZ value is larger than the SFLPAG value, can be rolled by pressing a Roll key without requiring any activity on the part of the program. CPF rolls the subfile unless the roll would go past the top or bottom of the subfile, in which case an error message is displayed. The DDS for this display is shown in Figure F-15.

NAME	DEPARTMENT	PHONE	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX	NNNN	+

The following numbered item refers to Figure F-15:

1 With the SFLSIZ value greater than the SFLPAG value, CPF can roll the subfile when the work station user presses a Roll Up or a Roll Down key. The program is unaware of the page being displayed. CPF rolls the subfile page by page (that is, by 17 records each time). For a description of rolling the subfile by page, see the ROLLUP/ROLLDOWN keyword description. For an example of rolling the subfile by other than the page value, see example 16.

IBM International Business Machines Corporation — **DATA DESCRIPTION SPECIFICATIONS** — GX21-7754-1 UM/050* Printed in U.S.A.

File | Programmer | Date | Keying Instruction | Graphic | Key | Description | Page of

A

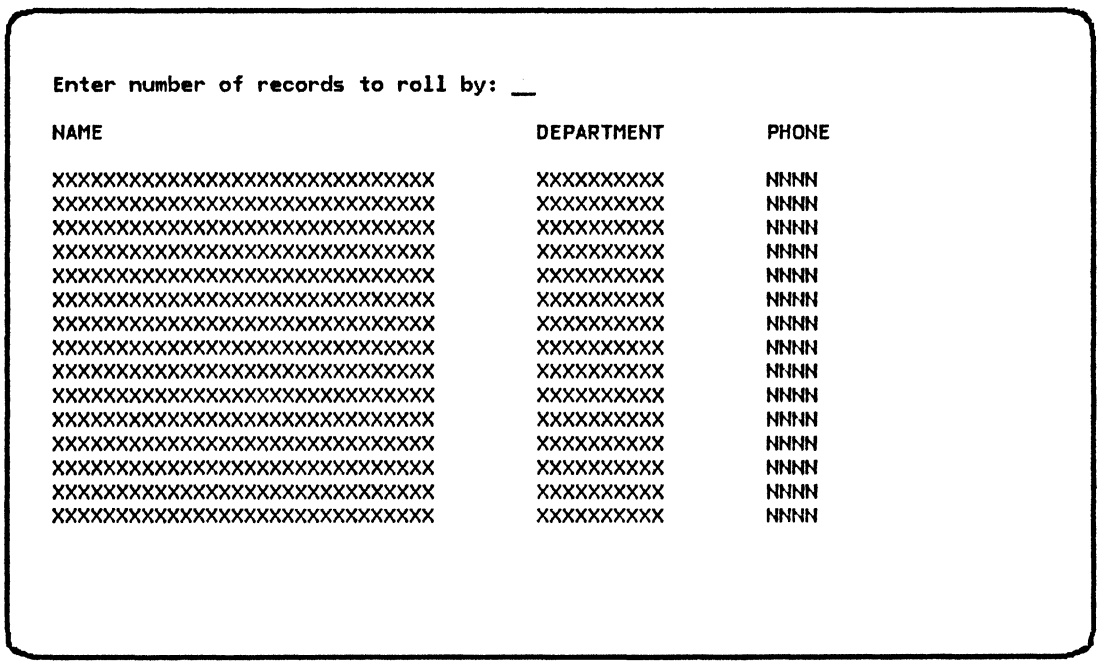
```
00010A*
00020A* SUBFILE WITH ROLLING BY CPF
00030A*
00040A          R SFLR                      SFL
00050A            NAME          30     5  2
00060A            DEPT          10     5 40
00070A            PHONE          4  0  5 58
00080A*
00090A          R SFLCTLR                   SFLCTL(SFLR)
00100A  60                                  SFLDSP
00110A  60                                  SFLDSPCTL
00120A                                      SFLSIZ(51)  } 1
00130A                                      SFLPAG(17)  }
00140A  30                                  SFLCLR
00150A                                 3  2'NAME'
00160A                                 3 40'DEPARTMENT'
00170A                                 3 58'PHONE'
```

*Number of sheets per pad may vary slightly.

Figure F-15. Subfile with Rolling by CPF

Example 16. Subfile with Rolling by CPF (SFLRCDNBR and SFLROLVAL Keywords)

The following display shows the subfile defined in Figure F-16. This example is similar to example 15, except that the work station user can key into a field to select how far to roll the subfile.

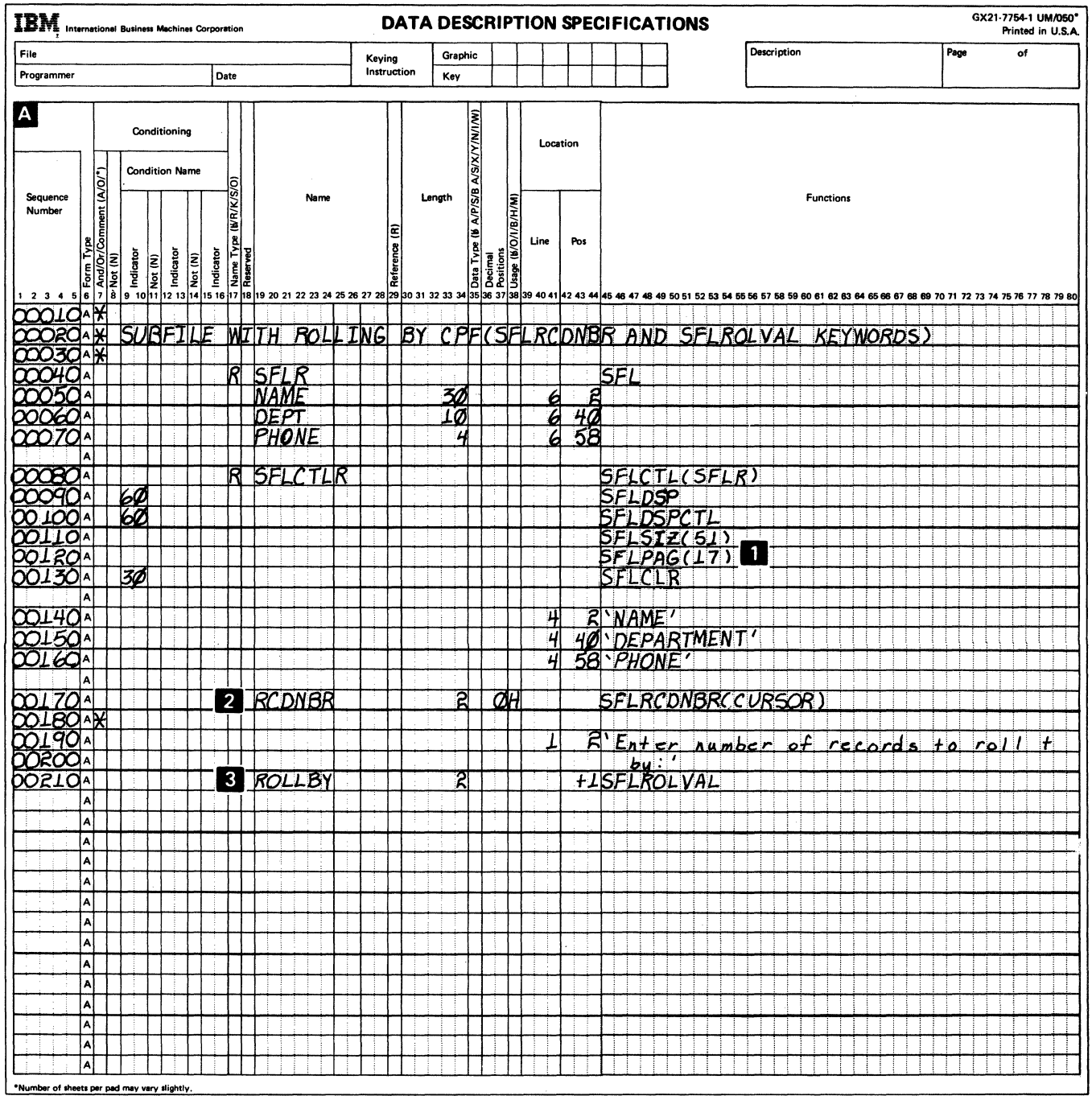


The following numbered items refer to Figure F-16:

1. When the work station user presses a Roll Up or a Roll Down key, the default is to roll one page of records (17 records for this subfile).

2. When the program displays the subfile, it first sets the value of RCDNBR to the relative record number of a subfile record to be displayed. When the subfile is first displayed, the page containing that subfile record is displayed and the cursor positioned at field NAME in the subfile record whose relative record number is in RCDNBR.

3. If the work station user keys a number into the field ROLLBY, then presses the Roll Up (or Roll Down) key, CPF adds (or subtracts) that number from the relative record number of the first record currently on the display, and displays the page of records beginning with the resulting relative record number.



IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050* Printed in U.S.A.

File | Programmer | Date | Keying Instruction | Graphic | Key | Description | Page | of

A

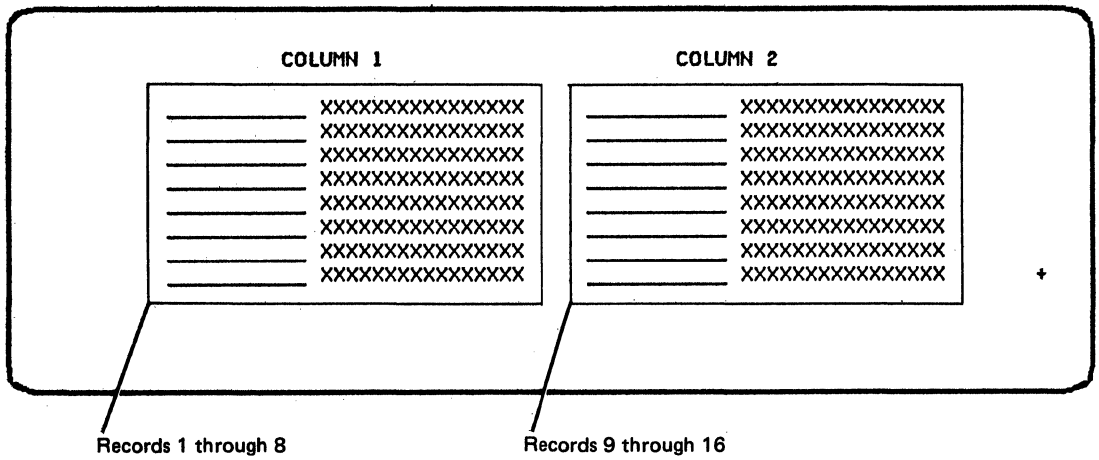
Sequence Number	Form Type	And/Or/Comment (A/O/*)	Not (N)	Conditioning: Condition Name Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (M/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A	*																	
00020	A	*		SUBFILE WITH ROLLING BY CPF(SFLRCDNBR AND SFLROLVAL KEYWORDS)															
00030	A	*																	
00040	A								R		SFLR								SFL
00050	A										NAME		30				6	2	
00060	A										DEPT		10				6	40	
00070	A										PHONE		4				6	58	
	A																		
00080	A								R		SFLCTLR								SFLCTL(SFLR)
00090	A			60															SFLDSP
00100	A			60															SFLDSPCTL
00110	A																		SFLSIZ(51)
00120	A																		SFLPAG(17) 1
00130	A			30															SFLCLR
	A																		
00140	A																4	2	'NAME'
00150	A																4	40	'DEPARTMENT'
00160	A																4	58	'PHONE'
	A																		
00170	A								2		RCDNBR		2		0H				SFLRCDNBR(CURSOR)
00180	A	*																	
00190	A																1	2	'Enter number of records to roll t
00200	A																		by:'
00210	A								3		ROLLBY		2						+1SFLROLVAL
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

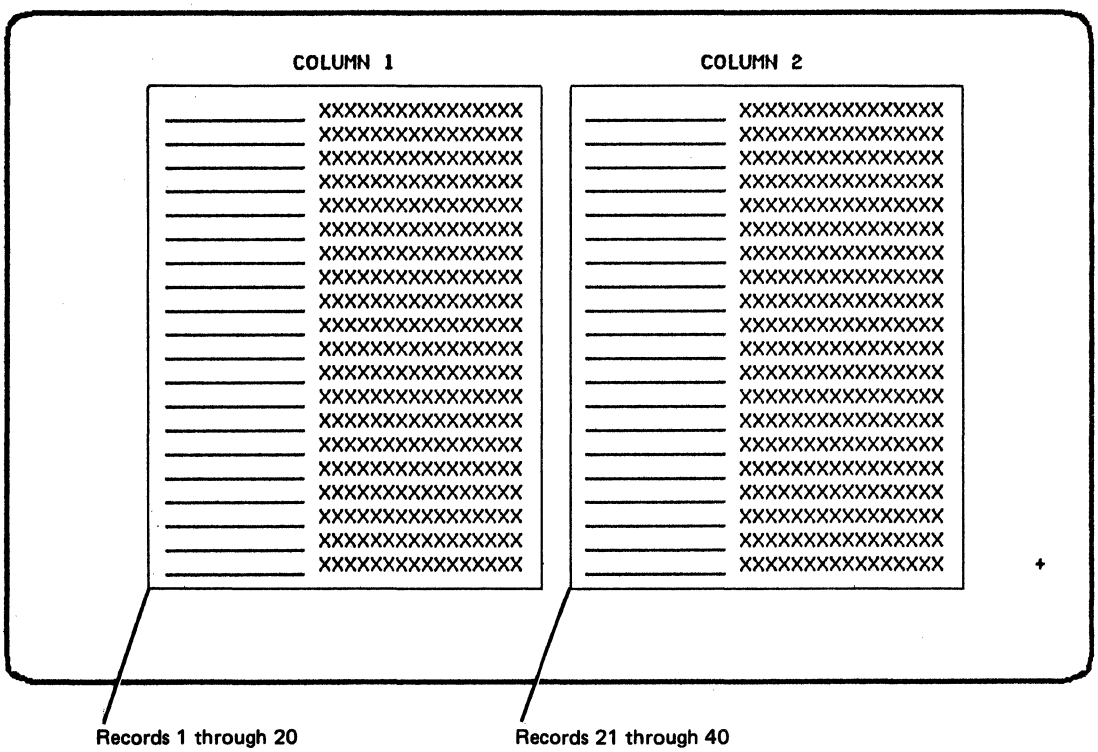
*Number of sheets per pad may vary slightly.

Figure F-16. Subfile with Rolling by CPF (SFLRCDNBR and SFLROLVAL Keywords)

Example 17. Horizontal Subfile That Is Also Displayable on Two Display Sizes

The following two displays show the subfile defined in Figure F-17 as it appears on the 12 by 80 and 24 by 80 screen sizes, respectively:





The following numbered items refer to Figure F-17:

1. There is one keyword at the file level, the keyword DSPSIZ (an optional keyword). This keyword has two values, *DS2 and *DS3, which state that the primary screen size is 12 lines by 80 characters and the secondary screen size is 24 lines by 80 characters. Lacking *DS1, this display file cannot be opened to the system console.

2. The SFLPAG keyword (a required keyword), is specified once with a value of 16 and again with a value of 40. The first time it applies to a device with the primary screen size (default of *DS2, or 12 lines by 80 characters); the second time, coded with a condition name of *DS3, it applies to a device with the secondary screen size (24 lines by 80 characters).

3. The SFLLIN keyword causes a subfile to be displayed horizontally. The parameter value specifies the number of spaces between columns of records. In this example, five spaces separate columns of records on both the 12 by 80 screen size (*DS2) and the 24 by 80 screen size (*DS3). Because *DS2 is the primary screen size, it does not need to be specified in positions 9 through 12.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050*
Printed in U.S.A.

File		Keying Instruction	Graphic	Description	Page	of
Programmer	Date		Key			

A

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (A/O/*) (7)	Conditioning: Not (N) / Indicator / Condition Name (8–16)	Name Type (b/R/K/S/O) (17)	Reserved (18)	Name (19–28)	Reference (R) (29)	Length (30–34)	Data Type (b A/P/S/B A/S/X/Y/N/I/W) (35)	Decimal Positions (36–37)	Usage (b/O/I/B/H/M) (38)	Location: Line (39–41)	Location: Pos (42–44)	Functions (45–80)
00010	A	* HORIZONTAL SUBFILE ON TWO SCREEN SIZES												
00020	A	*												
00030	A												**1**	DSPSIZ(*DS2 *DS3)
00040	A			R		SFL1								SFL
00050	A					FLDA		10	Y	0	I	3	11	
00060	A					FLDB		16			O	3	23	
	A													
00070	A			R		SFLCTL1								SFLCTL(SFL1)
00080	A													SFLSIZ(50)
00090	A												**2**	SFLPAG(16)
00100	A		*DS3											SFLPAG(38 40)
00110	A												**3**	SFLLIN(5)
00120	A		*DS3											SFLLIN(5)
	A													
00130	A		01											SFLEND
00140	A		02											SFLDSP
00150	A		03											SFLDSPCTL
00160	A		04											SFLCLR
	A													
00170	A											1	21	'COLUMN 1'
	A													
00180	A											1	55	'COLUMN 2'
	A													
	A													
	A													
	A													
	A													
	A													
	A													
	A													
	A													
	A													
	A													
	A													
	A													
	A													
	A													
	A													

*Number of sheets per pad may vary slightly.

Figure F-17. Horizontal Subfile on Two Screen Sizes

Example 18. Message Subfile

The following display shows the message subfile defined in Figure F-18:

```
                              MESSAGE SUBFILE

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX +
```

The following numbered items refer to Figure F-18:

1. Specifying the SFLMSGRCD keyword on the subfile record format identifies this subfile as a *message subfile*. The parameter value specified causes the subfile to be displayed on line 3 of the display.

2. The fields MSGKEY and PGMQ are user-defined names given to the two fields required for the subfile record format for a message subfile. The only specifications allowed for them are their names and the SFLMSGKEY and SFLPGMQ keywords, in the order shown.

   This subfile is built by a series of output operations to SFLR that place messages in the subfile as subfile records. Messages are truncated to fit single lines (76 characters or 61 characters, depending on screen size), and second-level help text is available. This subfile is displayed by an output operation to SFLCTLR with option indicator 01 set on.

3. This subfile is rolled by CPF when the work station user presses a Roll Up or a Roll Down key. The SFLEND keyword allows CPF to display a plus sign whenever the subfile can be rolled up.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050*
Printed in U.S.A.

File		Keying Instruction	Graphic		Description	Page	of
Programmer	Date		Key				

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning Indicators	Name Type	Name	Location Line	Location Pos	Functions
00010	A	*						
00020	A	*	MESSAGE SUBFILE					
00030	A	*						
00040	A			R	SFLR			SFL
00050	A							SFLMSGRCD(3) [1]
00060	A				MSGKEY [2]	[2]		SFLMSGKEY [2]
00070	A				PGMQ			SFLPGMQ
	A							
00080	A			R	SFLCTLR			SFLCTL(SFLR)
00090	A							SFLSIZ(12)
00100	A							SFLPAG(6)
00110	A		Ø1					SFLDSP [3]
00120	A		Ø2					SFLDSPCTL [3]
00130	A		Ø3					SFLCLR [3]
00140	A		Ø4					SFLEND [3]
	A							
	A							
00150	A					1	32	'MESSAGE SUBFILE'

*Number of sheets per pad may vary slightly.

Figure F-18. Message Subfile

Example 19. Display File with Three Subfiles

The following two displays show the three subfiles defined in Figure F-19. In the first display, two of the three subfiles are displayed. Subfile 1 is a two-column subfile showing what items a particular account has on order. Subfile 2 is a single-column subfile showing which customers have ordered a particular item in stock.

This display might be used by a shipping clerk who is allocating in-stock items to customer orders. As the clerk allocates the items, he would enter the amounts to be shipped into the QTY SHIPD input fields. If the in-stock amount drops below a certain amount, the message subfile could be displayed.

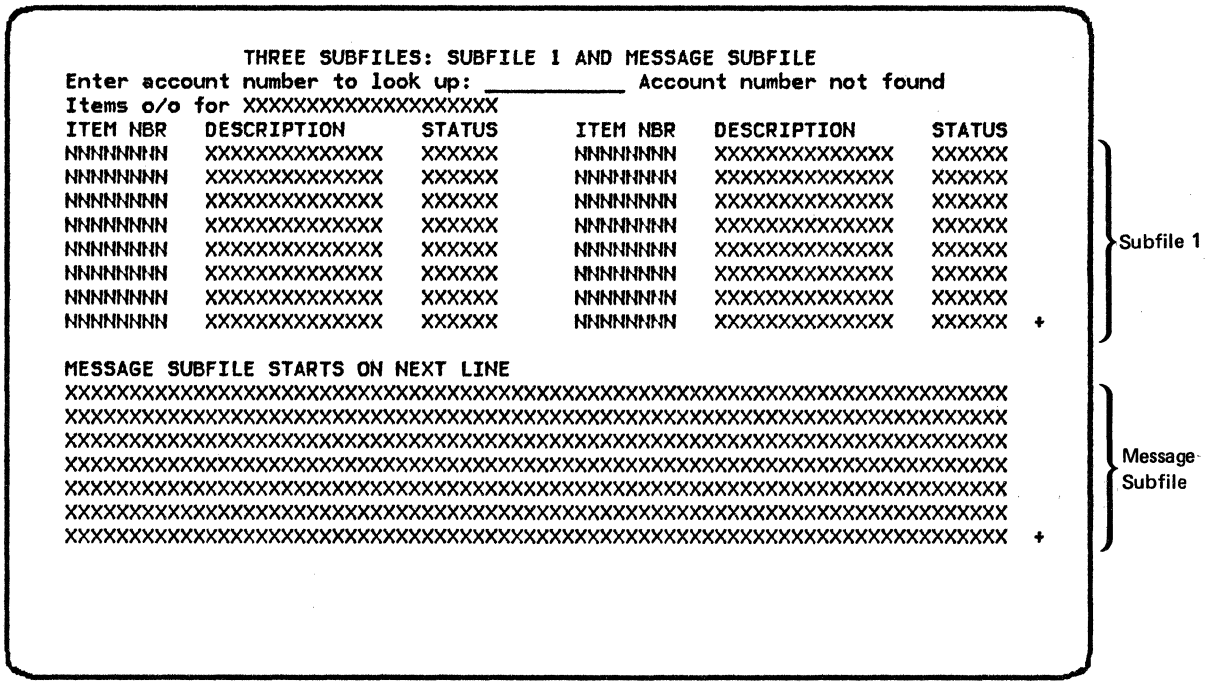
```
                    THREE SUBFILES: SUBFILE 1 AND SUBFILE 2
Enter account number to look up: __________ Account number not found
Items o/o for XXXXXXXXXXXXXXXXXXXX
ITEM NBR   DESCRIPTION       STATUS       ITEM NBR   DESCRIPTION       STATUS
NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX       NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX      }
NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX       NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX      }
NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX       NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX      }
NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX       NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX      } Subfile 1
NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX       NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX      }
NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX       NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX      }
NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX       NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX      }
NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX       NNNNNNNN   XXXXXXXXXXXXXX    XXXXXX  +   }

Enter item number: ________
Customers that have ordered item XXXXXXXXXXXXXX
CUSTOMER NAME              ACCOUNT NBR   DATE ORDERED    QTY O/O    QTY SHIPD
XXXXXXXXXXXXXXXXXXXXX      NNNNNNNNNNN   NN/NN/NN        NNNNNN     _______        }
XXXXXXXXXXXXXXXXXXXXX      NNNNNNNNNNN   NN/NN/NN        NNNNNN     _______        }
XXXXXXXXXXXXXXXXXXXXX      NNNNNNNNNNN   NN/NN/NN        NNNNNN     _______        } Subfile 2
XXXXXXXXXXXXXXXXXXXXX      NNNNNNNNNNN   NN/NN/NN        NNNNNN     _______        }
XXXXXXXXXXXXXXXXXXXXX      NNNNNNNNNNN   NN/NN/NN        NNNNNN     _______    +   }

CF1 - End program   CF2 - Look up new account   CF3 - Look up new item
Item number not found
```

In the second display the message subfile displays messages pertaining to the items the shipping clerk has tried to allocate to the order being filled, such as when items were back-ordered and expected arrival dates.

Note: Subfile 2 must be deleted before the message subfile is built; it is the responsibility of the program to save any important data before deleting a subfile.



The following numbered items refer to part 1 of Figure F-19:

1. The program can select which command keys are in effect by setting option indicator 50. The CF and CA keywords are specified at the file level, so the response indicators specified with the CF and CA keywords are returned to the program on any input operation issued to any record format.

2. Subfile 1 is a horizontally displayed subfile; there are 16 records in a page, displayed in two columns of eight records each, with six spaces between columns. There are 64 records in the entire subfile.

3. Specifying the OVERLAY keyword in the subfile control record format permits the program to display subfile 1 without erasing other records on the display.

4. If the account number entered into the fields ACCTNBR1 is invalid, the program can redisplay the field with its image reversed and can position the cursor at the field. To do this, the program issues an output operation to SFLCTL1 with option indicators 02 and 91 set on. (The constant field 'Account number not found' is displayed as a highlighted field.)

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050* Printed in U.S.A.

File		Keying Instruction	Graphic								Description	Page of
Programmer	Date		Key									

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B/A/S/K/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
00010	A	*																	
00020	A	*	THREE SUBFILES IN ONE DISPLAY FILE:																SUBFILE 1
00030	A	*																	
00040	A																		CF01(21 'End program')
00050	A																		CF02(22 'Look up new account')
00060	A			50													1		CF03(23 'Look up new item')
00070	A		N50																CF08(28 'Delete message subfile, t
00080	A																		then display subfile 1')
00090	A	*																	
00100	A								R		SFL1								SFL
00110	A										ITMNBR		8	Y	0		5	2	
00120	A										DESCR1		14					+3	
00130	A										STATUS		6			B		+3	
00140	A	*																	
00150	A								R		SFLCTL1								SFLCTL(SFL1)
00160	A																		SFLSIZ(64) SFLPAG(16) SFLLIN(6) 2
00170	A			01															SFLDSP
00180	A			02															SFLDSPCTL
00190	A			03															SFLCLR
00200	A			04															SFLEND
00210	A																		OVERLAY 3
00220	A	*																	
00230	A																2	2	'Enter account number to look up:'
00240	A										ACCTNBR1		11	Y	0B			+1	CHECK(MF)
00250	A			91															DSPATR(RI PC) 4
00260	A			91														+1	'Account number not found'
00270	A																		DSPATR(HI)
00280	A																3	2	'Items o/e for'
00290	A										ACCTNME		20					+1	
00300	A																4	2	'ITEM NBR'
00310	A																	+3	'DESCRIPTION'
00320	A																	+6	'STATUS'
00330	A																	+6	'ITEM NBR'
00340	A																	+3	'DESCRIPTION'
00350	A																	+6	'STATUS'
	A																		
	A																		
	A																		

*Number of sheets per pad may vary slightly.

Figure F-19 (Part 1 of 4). Three Subfiles in One Display File

The following numbered items refer to part 2 of Figure F-19:

5 Subfile 2 is a vertically displayed subfile; there are five records in a page, displayed in one column (the SFLLIN keyword is not specified). There are 25 records in the entire subfile.

6 Specifying the SFLDLT keyword with an option indicator enables the program to delete this subfile when a third subfile is to be built and displayed. To delete the subfile, the program issues an output operation to SFLCTL2 with option indicator 05 set on.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050* Printed in U.S.A.

File | Programmer | Date | Keying Instruction | Graphic | Key | Description | Page | of

A

Sequence Number	Form Type	Conditioning: And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Condition Name: Not (N)	Condition Name: Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B/A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A	*																	
00020	A	*		THREE SUBFILES IN ONE DISPLAY FILE: SUBFILE 2															
00030	A	*																	
00040	A								R		SFL2								SFL
00050	A										CUST		20				17	2	
00060	A										ACCTNBR2		11	Y	0			+3	
00070	A										DATORD		6		0			+3	EDTCDE(Y)
00080	A										QTY00		6	Y	0			+7	
00090	A										QTYSHP		6	Y	0	I		+4	CHECK(RB) COMP(GE 0)
00100	A	*																	
00110	A								R		SFLCTL2								SFLCTL(SFL2)
00120	A																		SFLSIZ(25) } 5
00130	A																		SFLPAG(5) }
00140	A			01															SFLDSP
00150	A			02															SFLDSPCTL
00160	A			03															SFLCLR
00170	A			04															SFLEND
00180	A			05														6	SFLDLT
00190	A																		OVERLAY
00200	A	*																	
00210	A																14	2	'Enter item number:'
00220	A										ITMNBR		8	Y	0	B		+1	CHECK(MF)
00230	A			91															ERRMSG('Item number not found')
00240	A																15	2	'Customers that have ordered item'
00250	A										DESCR2		14					+1	
00260	A																16	2	'CUSTOMER NAME +'
00270	A																		ACCOUNT NBR +
00280	A																		DATE ORDERED +
00290	A																		QTY O/O +
00300	A																		QTY SHIPD'
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

*Number of sheets per pad may vary slightly.

Figure F-19 (Part 2 of 4). Three Subfiles in One Display File

The following numbered items refer to part 3 of Figure F-19:

7 Specifying SFLMSGRCD in the subfile record format for SFL3 makes SFL3 a message subfile. The SFLMSGKEY and SFLPGMQ keywords are required on the two fields specified in the subfile record format of a message subfile. These two fields are names MSGKEY and PGMQ in this example.

8 The SFLDLT keyword permits the program to delete this subfile after it has been built and displayed by issuing an output operation to SFLCTL3 with option indicator 05 set on.

9 The program can build this message subfile in one of two ways by setting option indicators:

- A *multiple output operation* (issued to SFLCTL3) places several records in the subfile with one output operation. For this example, option indicator 08 must be set on for the multiple output operation.
- The program can build the message subfile with as many *single output operations* as required by first setting off option indicator 08. Then, the program can determine the value of the fields MSGKEY and PGMQ for each output operation to SFL3.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050* Printed in U.S.A.

File | Programmer | Date | Keying Instruction | Graphic | Key | Description | Page of

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Not (N)	Condition Name: Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blR/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (b A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (b/O/I/B/H/M)	Location: Line	Pos	Functions
00010	A	X																	
00020	A	X		THREE SUBFILES IN ONE DISPLAY FILE:							MESSAGE SUBFILE								
00030	A	X																	
00040	A								R		SFL3								SFL SFLMSGRCD(15) 7
00050	A										MSGKEY								SFLMSGKEY
00060	A										PGMQ								SFLPGMQ
00070	A	X																	
00080	A								R		SFLCTL3								SFLCTL(SFL3)
00090	A																		SFLSIZ(15)
00100	A																		SFLPAG(7)
00110	A			01															SFLDSP
00120	A			02															SFLDSPCTL
00130	A			03															SFLCLR
00140	A			04															SFLEND
00150	A			05															SFLDLT 8
00160	A																		OVERLAY
00170	A			08															SFLINZ 9
00180	A										PGMQ								SFLPGMQ
00190	A										RCDNBR		2	0	H				SFLRCDNBR(CURSOR)
00200	A	X																	
00210	A																14	2	'MESSAGE SUBFILE STARTS ON NEXT +
00220	A																		LINE'

*Number of sheets per pad may vary slightly.

Figure F-19 (Part 3 of 4). Three Subfiles in One Display File

The following numbered items refer to part 4 of Figure F-19:

10 To change the title to match the displayed subfiles, the program sets indicator 50 as required and issues an output operation to TITLES.

11 To change the instructions displayed at the bottom of the screen when displaying the message subfile or replacing it with subfile 2, the program sets option indicator 50 as required and issues an output operation to KEYS. (Setting the indicator also affects the command keys specified at the file level; see part 1 of Figure F-19.)

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050* Printed in U.S.A.

File		Keying Instruction	Graphic								Description	Page	of
Programmer	Date		Key										

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning: Condition Name: Not (N)	Indicator	Not (N)	Indicator	Not (N)	Indicator	Name Type (blank/R/K/S/O)	Reserved	Name	Reference (R)	Length	Data Type (blank/A/P/S/B A/S/X/Y/N/I/W)	Decimal Positions	Usage (blank/I/O/I/B/H/M)	Location: Line	Location: Pos	Functions
1 2 3 4 5	6	7	8	9 10	11	12 13	14	15 16	17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
00010	A	X																	
00020	A	X		HOUSEKEEPING RECORD FORMATS															
00030	A	X																	
00040	A									R	TITLES								TEXT('For subfile R, set ind 50 on +
00050	A																		for message subfile, set ind +
00060	A																		50 off') OVERLAY
00070	A		50														1	20	'THREE SUBFILES: SUBFILE 1 +
00080	A																		AND SUBFILE R'
00090	A		N50														1	17	'THREE SUBFILES: SUBFILE R +
00100	A																		AND MESSAGE SUBFILE'
00110	A	X																	
00120	A									R	KEYS								OVERLAY
00130	A		50														23	2	'CF1 - End program CF2 - Look +
00140	A																		up new account +
00150	A																		CF3 - Look up new item +
00160	A		N50																CF8 - Remove messages'

*Number of sheets per pad may vary slightly.

Figure F-19 (Part 4 of 4). Three Subfiles in One Display File

Example 20. Printer File

The following printer file contains DDS for printing a customer master list.

The following numbered items apply to part 1 of Figure F-20:

1. This printer file references the field reference file MLGREFP.

2. When SKIPB(2) is specified at the record level, the printer skips to line 2 before printing the record format (HEADER). Also, line numbers in positions 39 through 41 are invalid.

3. UNDERLINE is a field level keyword that causes the constant field preceding it to be underlined on the printout.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050*
Printed in U.S.A.

File		Keying Instruction	Graphic		Description	Page	of
Programmer	Date		Key				

```
00010A*
00020A* SAMPLE PRINTER FILE
00030A                                     1  REF(MLGREFP)
00040A          R HEADER                      TEXT('TWO-LINE HEADING, UNDERLINED')
00050A                                     2  SKIPB(2)
00060A                                      29'CUSTOMER MASTER FILE'
00070A                                      75DATE   EDTCDE(Y)
00080A                                      +1TIME
00090A                                     122'Page'
00100A                                      +1PAGNBR  EDTCDE(Z)  SPACEA(2)
00110A                                       2'ACCOUNT  CUSTOMER'
00120A                                        SPACEA(1)
00130A                                       2'NUMBER   NAME                      +
00140A                                        ADDRESS                +
00150A                                        CITY                   +
00160A                                        STATE    ZIP  '
00170A                                     3  UNDERLINE
00180A                                        SPACEA(2)
```

*Number of sheets per pad may vary slightly.

Figure F-20 (Part 1 of 2). Printer File

The following numbered item applies to part 2 of Figure F-20:

4 The REFFLD keyword specifies that ACTNBR is to have the same attributes as ACTNUM; ACTNUM is defined in file MLGREFP as specified on the REF keyword in part 1 of this figure.

IBM International Business Machines Corporation — **DATA DESCRIPTION SPECIFICATIONS** — GX21-7754-1 UM/050* Printed in U.S.A.

File | Keying Instruction | Graphic | Key | Programmer | Date | Description | Page of

A

Sequence Number (1–5)	Form Type (6)	And/Or/Comment (7)	Conditioning (8–16)	Name Type (17)	Reserved (18)	Name (19–28)	Reference (29)	Length (30–34)	Data Type (35)	Decimal Positions (36–37)	Usage (38)	Location Line (39–41)	Location Pos (42–44)	Functions (45–80)
00010	A			R		DETAIL								TEXT('ONE-LINE RECORD')
00020	A													SPACEA(1)
00030	A		**4**			ACTNBR	R						2	REFFLD(ACTNUM)
00040	A					NAME	R						+4	
00050	A					ADDRESS	R						+3	
00060	A					CITY	R						+3	
00070	A					STATE	R						+3	
00080	A					ZIP	R						+6	

*Number of sheets per pad may vary slightly.

Figure F-20 (Part 2 of 2). Printer File

Example 21. Communications File or BSC File

The following communications file or BSC file contains DDS for transmitting data between System/38 and a remote system or device.

The following numbered items apply to Figure F-21:

1. Two record formats are specified for this file. In the send state, INQ is used; in the receive state, ANS is used.

2. When your program can no longer continue a session, it selects the FAIL keyword by setting on option indicator 15 and issuing an output operation.

3. When option indicator 10 is set on, the ALWWRT keyword allows CPF to inform the remote system or device that the program has finished sending.

4. When option indicator 11 is set on, the RQSWRT keyword allows CPF to request permission to send data.

5. The RCVENDGRP keyword allows response indicator 13 to be set on at the end of a user-defined group of records.

6. The TRNRND keyword allows response indicator 14 to be set on when the remote system or device has stopped sending.

For more information on how to use communications files and BSC files, see the *Data Communications Programmer's Guide*.

IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754-1 UM/050*
Printed in U.S.A.

File | Keying Instruction | Graphic | Key | Description | Page of
Programmer | Date

A

Sequence Number	Form Type	And/Or/Comment (A/O/*)	Conditioning Indicator	Name Type (b/R/K/S/O)	Name	Reference (R)	Length	Functions
00010	A	*						
00020	A	* SAMPLE COMMUNICATIONS FILE						
00030	A	*						
00040	A							REF(MLGREFP) [2]
00050	A		15					FAIL
00060	A			R [1]	INQ			TEXT('Inquiry from System/38')
00070	A		10					ALWWRT [3]
00080	A		11					RQSWRT [4]
00090	A							
00100	A				TRAN		5	
00110	A				CUSTP		10	
00120	A							
00130	A			R [1]	ANS			TEXT('Answer from remote system +
00140	A							or device')
00150	A							RCVENDGRP(13) [5]
00160	A							TRNRND(14) [6]
00170	A				ACTNUM	R		
00180	A				NAME	R		

*Number of sheets per pad may vary slightly.

Figure F-21. Communications or BSC File

File Specifications in RPG

The following RPG specifications are shown only in part to illustrate how externally described data is specified in an RPG program.

The following numbered items apply to Figure F-22:

1. The E in position 19 of the file specifications refers to the external descriptions (the DDS).

2. The DDS from the logical data base file CUSMSTL and from the display file CUSRVWD would be included in the output of the compilation of this program.

3. The relative record field RECNUM is used in the program to read and write records to the subfile.

4. The subfile record format name is SFLR.

IBM International Business Machines Corporation

RPG CONTROL AND FILE DESCRIPTION SPECIFICATIONS

GX21-9092-7 UM/050*
Printed in U.S.A.

Program		Keying Instruction	Graphic	Card Electro Number	Page of	Program Identification (75–80)
Programmer	Date		Key			CUSRVW

Control Specifications

For the valid entries for a system, refer to the RPG reference manual for that system.

Line (3–5)	Form Type (6)
01	H

File Description Specifications

For the valid entries for a system, refer to the RPG reference manual for that system.

Line (3–5)	Form Type (6)	Filename (7–14)	File Type (15)	File Designation (16)	File Format (19)	Record Address Type (31)	Device (40–46)	Symbolic Device / Continuation Lines (47–52)	K (53)	Option (54–59)	Entry (60–65)
02	F	** CUSTOMER REVIEW									
03	F	*									
04	F	CUSMSTL	I	F	E	K	DISK				
05	F	CUSRVWD	C	F	E		WORKSTN				
06	F							RECNUM	K	SFILE	SFLR
07	F										
08	F										
09	F										
10	F										
	F										
	F										

Callouts: 1 (File Format E, lines 04–05), 2 (Filename, line 05), 3 (positions 47–52, line 06), 4 (Entry, line 06).

*Number of sheets per pad may vary slightly.

Figure F-22. A Partial RPG Program Showing How to Specify Externally Described Data

The following partial COBOL program illustrates how externally described data is specified in a COBOL program.

The following numbered items apply to Figure F-23:

1. The COBOL program refers to file CUSMSTL as CUST-MASTER.

2. The COPY verb specifies that the record format CUSREC of logical file CUSMSTL is to supply the data descriptions for file CUSMSTL. The COBOL program refers to CUSREC as CUST-REC.

IBM

COBOL Coding Form

SYSTEM S/38		PUNCHING INSTRUCTIONS
PROGRAM		GRAPHIC
PROGRAMMER	DATE	PUNCH

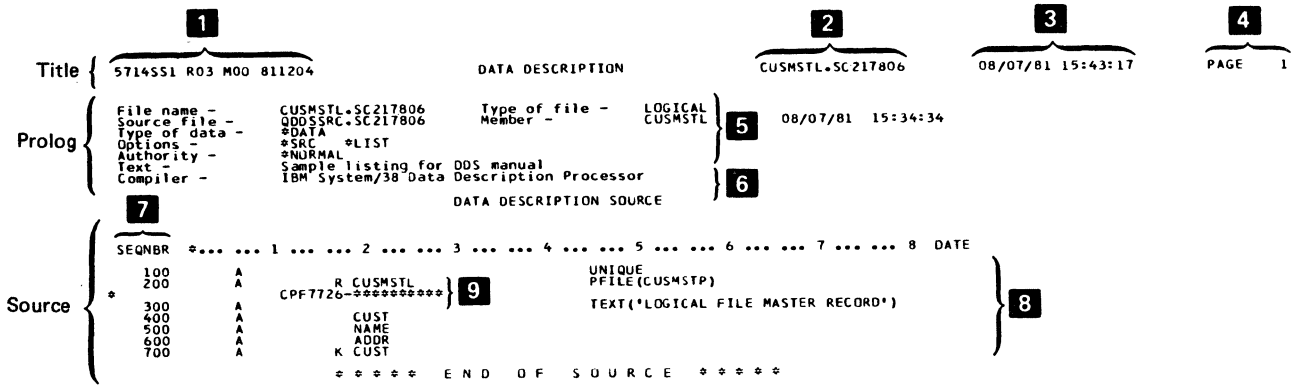
SEQUENCE (PAGE)	SEQUENCE (SERIAL)	CONT.	A	B	COBOL STATEMENT
	01				
	02				
0000	03				DATA DIVISION.
0000	04				FILE SECTION.
0000	05				FD CUST-MASTER
0000	06				LABEL RECORDS ARE STANDARD.
0000	07				01 CUST-REC.
0000	08				COPY DDS-CUSREC OF CUSMSTL.
	09				
	10				
	11				
	12				
	13				

Figure F-23. A Partial COBOL Program Showing How to Specify Externally Described Data

SAMPLE COMPILER LISTING

Once data description specifications are written, they must be put into a source file. Then, data base or device files are created by entering the CL command that invokes the data description processor. The CL command can be entered interactively or in a batch job. The data description processor retrieves the data description specifications from the source file designated on the create file command, validates the specifications, and creates an output listing with any errors and any referenced specifications.

A sample data description specifications compiler listing follows:



Title (appears at top of each output page):

1. The program number, release modification level, and date of CPF.
2. The qualified object name.
3. The date and time of this run.
4. The page number in the listing.

Prolog:

5. The type of file and the parameter values specified (or defaults if not specified) on the create file command.
6. The name of the DDS processor.

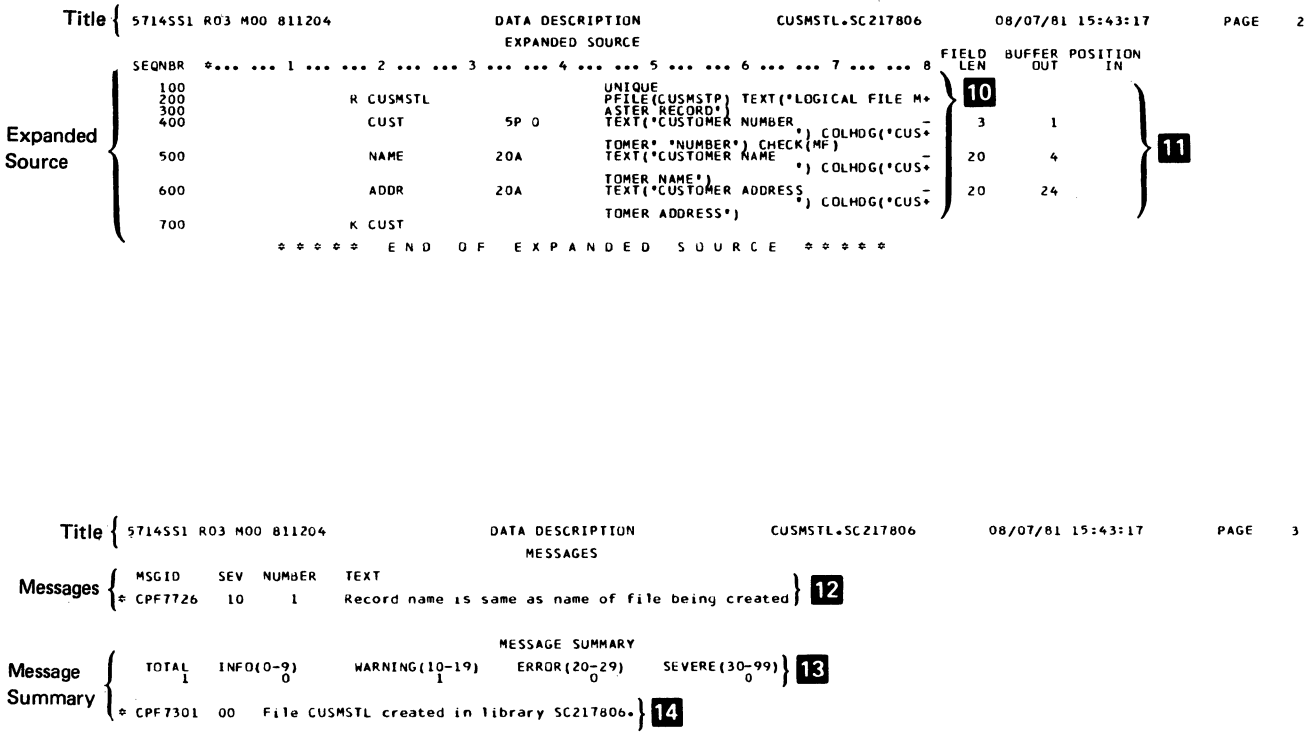
Source:

7. The sequence numbers of lines (records) in the source. Comments are treated like any other specification line and are given sequence numbers.

   See the chapter on *Source Files* in the *CPF Programmer's Guide* for how sequence numbers are assigned.

8. The source specifications.
9. If an error is found during processing of the DDS and can be traced specifically to a source specification, the error message identifier and an asterisk indicating where the error is are printed immediately following the source specification line. An asterisk is also printed under the sequence number to indicate that the line contains an error message.

   For more information about DDS processing errors, see the section on *Errors Encountered During DDS Processing* in the chapter on *Data Base* in the *CPF Programmer's Guide*.


Title
5714SS1 R03 M00 811204 DATA DESCRIPTION CUSMSTL.SC217806 08/07/81 15:43:17 PAGE 2
EXPANDED SOURCE
Expanded Source
SEQNBR *... ... 1 ... ... 2 ... ... 3 ... ... 4 ... ... 5 ... ... 6 ... ... 7 8
FIELD LEN
BUFFER OUT
POSITION IN
100 UNIQUE
200 R CUSMSTL PFILE(CUSMSTP) TEXT('LOGICAL FILE M+
300 ASTER RECORD')
400 CUST 5P 0 TEXT('CUSTOMER NUMBER -
') COLHDG('CUS+
3 1
TOMER' 'NUMBER') CHECK(MF)
500 NAME 20A TEXT('CUSTOMER NAME -
') COLHDG('CUS+
20 4
TOMER NAME')
600 ADDR 20A TEXT('CUSTOMER ADDRESS -
') COLHDG('CUS+
20 24
TOMER ADDRESS')
700 K CUST
* * * * * E N D O F E X P A N D E D S O U R C E * * * * *
10
11

Title
5714SS1 R03 M00 811204 DATA DESCRIPTION CUSMSTL.SC217806 08/07/81 15:43:17 PAGE 3
MESSAGES
Messages
MSGID SEV NUMBER TEXT
* CPF7726 10 1 Record name is same as name of file being created
12
MESSAGE SUMMARY
Message Summary
TOTAL 1
INFO(0-9) 0
WARNING(10-19) 1
ERROR(20-29) 0
SEVERE(30-99) 0
13
* CPF7301 00 File CUSMSTL created in library SC217806.
14

Expanded Source:

10 Only the valid DDS. This list is what is actually in the file description. No comments or messages are printed. Default values and referenced values are printed for the valid DDS.

11 The length and the buffer (input or output) position of each field.

Messages:

12 This section contains a list of all messages (general messages and those already indicated in the source section) encountered during processing of the DDS. For each message, the message identifier, the severity, the number of times the message occurred, and the message text are listed.

Message Summary:

13 The number of messages at each severity level.

14 The final completion message.

DEBUGGING TEMPLATE

A special template is available to aid you in interpreting the fields on the DDS compiler listing. Figure F-24 shows a reduced *(DDS) Debugging Template,* GX21-7717-0.

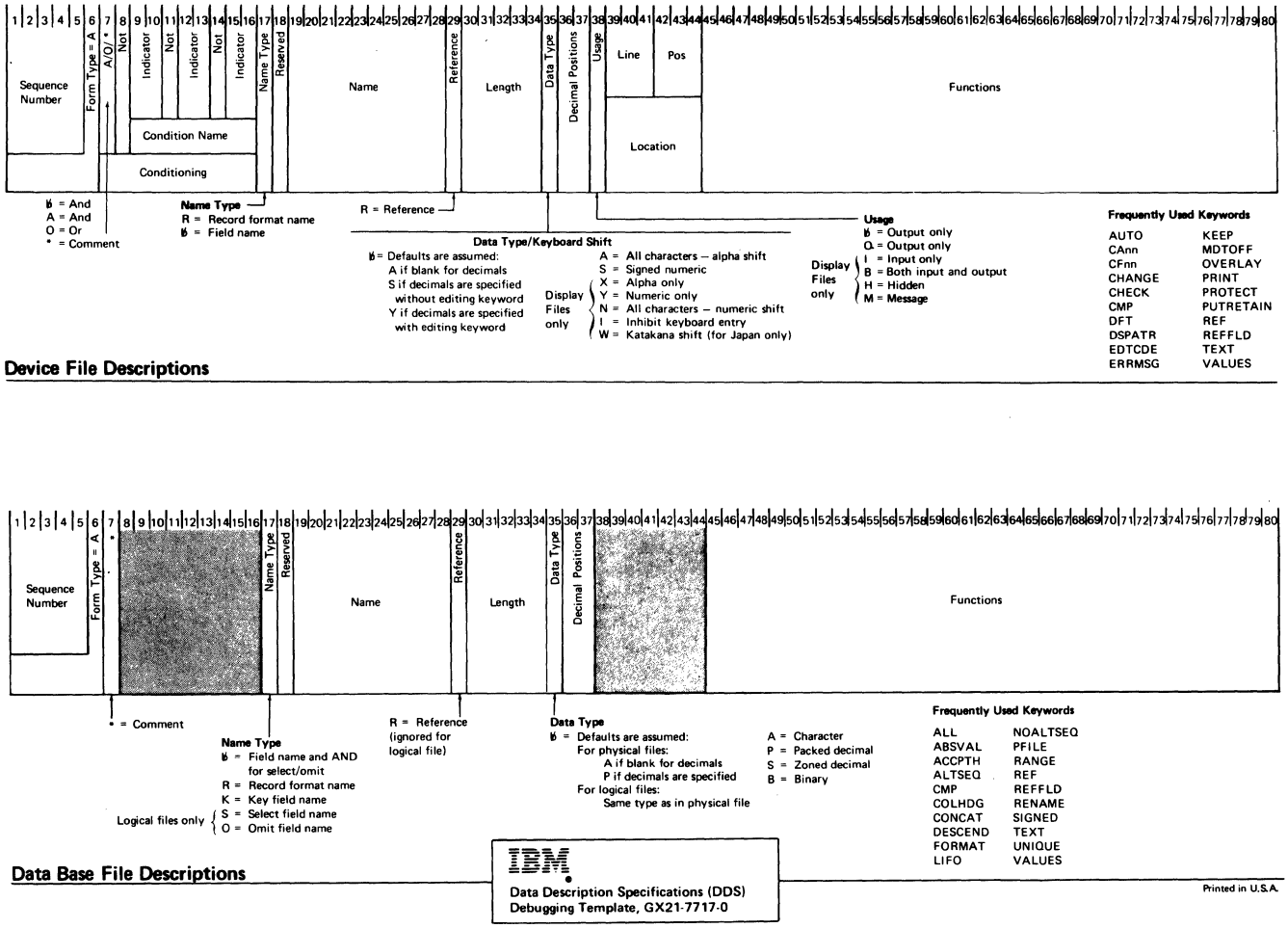
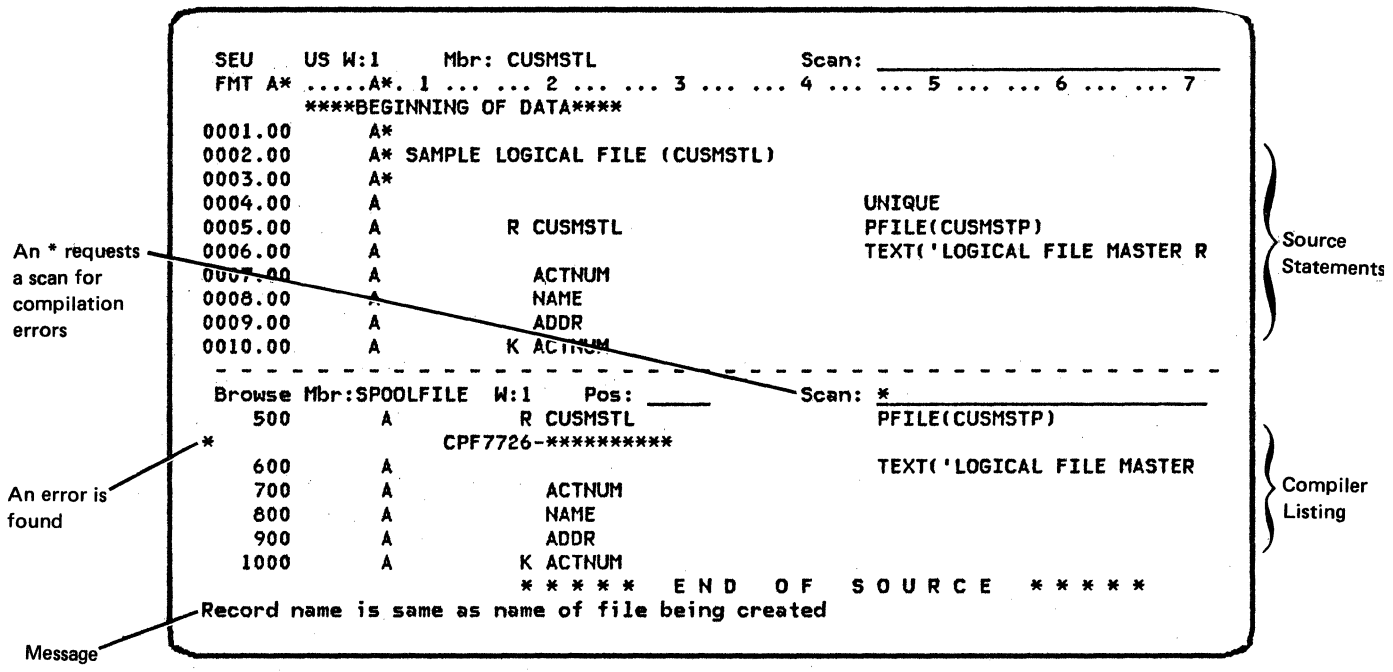


Figure F-24. IBM Data Description Specifications Debugging Template (reduced)

USING SEU TO BROWSE THROUGH A COMPILER LISTING AND UPDATE SOURCE STATEMENTS

When your compiler listing is on an ouput queue, you can use SEU to browse through (view) your listing.

The following shows the split-edit display available in SEU, which allows you to browse through your compiler listing at your work station:



While browsing through the listing, you can:

- Scan for errors
- Press the Help key to display second-level text for error messages
- Correct the source statements that have errors

For detailed information about browsing through a compiler listing, see the *SEU Reference Manual*.

Appendix G. Data Description Specifications Keyword and Value Abbreviations

The following list contains all keyword and value abbreviations used in data description specifications. Each abbreviation is listed in alphabetical order, followed by its associated meaning.

A	after
AB	allow blanks
ABS	absolute
ACC	access
ALT	alternate
ALW	allow
ATR	attributes
AUTO	automatic
B	before
BL	blinking field
BLANKS	blanks
BLINK	blinking cursor
BLK	blank
CA	command attention key
CDE	code
CF	command function key
CHG	change
CHK	check
CLR	clear
CLRL	clear line
CLS	class
CMD	command
CMP	comparison
COL	column
COMP	comparison
CON	constant
CONCAT	concatenate
CS	column separator
CSR	cursor
CTL	control
DEV	device
DFN	defined
DFT	default
DLT	delete
DSP	display
DTA	data
DUP	duplicate
EDT	edit
END	end
EQ	equal
ER	end of record (= end of field)
ERR	error
FAIL	fail
FE	field exit
FLD	field
FMT	format
GE	greater than or equal to
GRP	group
GT	greater than
HDG	heading
HI	high intensity
ID	identifier
IND	indicator
INP	input
INZ	initialize
LC	lowercase
LE	less than or equal to
LIFO	last in first out
LIN	line
LOC	location
LT	less than
MDT	modified data tag
ME	mandatory enter
MF	mandatory fill

MSG	message
M10	IBM Modulus 10
M11	IBM Modulus 11
NBR	number
ND	nondisplay
NE	not equal to
NG	not greater than
NL	not less than
NXT	next
OF	off
OID	operator identification
OUT	output
OVR	override
P	physical
PAG	page
PC	position cursor
PFILE	physical file
PGM	program
POS	position
PR	protect
PRT	print or printer
PTH	path
Q	queue
RA	record advance
RAB	right-adjust with blank fill
RAZ	right-adjust with zero fill
RB	right-adjust with blank fill
RCD	record
RCV	receive
REF	reference
RI	reverse image
RL	right to left
RLTB	right to left, top to bottom
RNA	record not active
ROL	roll
RQS	request
RTG	routing
RTN	return
RZ	right-adjust with zero fill
SEQ	sequence
SFL	subfile
SIZ	size
SKIPA	skip after
SKIPB	skip before
SLNO	starting line number
SP	select by light pen
SPACEA	space after
SPACEB	space before
TRNRND	turn around
TXT	text
UL	underline
USR	user
VAL	values
VLD	valid
VN	valid name
WRD	word
WRT	write

Appendix H. Glossary

This glossary is a subset of the System/38 glossary. The System/38 glossary defines terms that are used in the customer documentation for System/38.

This glossary includes definitions developed by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO). This material is reproduced from the *American National Dictionary for Information Processing*, copyright 1977 by the Computer and Business Equipment Manufacturers Association. Copies of this dictionary may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions from this dictionary are identified by (ANSI) at the beginning of the definition.

access path: The means by which CPF provides a logical organization to the data in a data base file so that the data can be processed by a program. See also *arrival sequence access path* and *keyed sequence access path.*

acknowledgment character: In BSC, a transmission control character that is sent as a positive response to a data transmission.

ACK0: In BSC, the even-numbered positive acknowledgment character. The ACK0 character indicates that text was received without transmission errors. See *acknowledgment character.*

ACK1: In BSC, the odd-numbered positive acknowledgment character. The ACK1 character indicates that text was received without transmission errors. See *acknowledgment character.*

alphabetic character: (1) Any one of the letters A through Z (uppercase and lowercase) or one of the characters #, $, or @. (2) In COBOL, a character that is one of the 26 uppercase characters of the alphabet, or a space.

alphameric character: Any one of the alphabetic characters, one of the digits 0 through 9, or the character __ (underscore) as defined in CPF.

alphanumeric character: In COBOL, any character in the computer's character set.

alternate collating sequence: A collating sequence that differs from the normal collating sequence or that allows two or more characters to be considered equal. See *collating sequence.*

AND relationship: The specification of conditioning indicators so that the operation is performed only when all conditions are met.

application: (1) A particular data processing task, such as an inventory control application or a payroll application. (2) In IDU, specialized program created by IDU from user input. An application is later called by DFU or the query utility.

application program: A program used to perform a particular data processing task such as inventory control or payroll.

arrival sequence access path: An access path that is based on the order in which records are stored in a physical file. Contrast with *keyed sequence access path.*

ascending key sequence: The arrangement of data in an order from the lowest value of the key field to the highest value of the key field. Contrast with *descending key sequence.*

ascending sequence: The arrangement of data in an order from low to high based on the contents of a specific field or fields. Contrast with *descending sequence.*

attribute: A characteristic; for example, attributes of a field include its length and data type, and attributes of a job include its user name and job date.

attribute character: A character associated with a field in a display file that defines how the field is displayed (such as underlined, blinking, or intensified).

authority: The right to access objects, resources, or functions.

batch device: Any device that can read serial input or write serial output, or both, but cannot communicate interactively with the system. Examples of batch devices are card devices, printers, and diskette units.

batch job: A group of processing actions submitted as a predefined series of actions to be performed with little or no interaction between the user and the system.

batch processing: A method of executing a program or a series of programs in which one or more records (a batch) is processed with little or no interaction with the user or operator. Contrast with *interactive processing.*

beginning attribute character: For a display file, the character that precedes the first position in a field and that defines the displayed field.

binary: Relating to, being, or belonging to a numbering system with a base of 2. Valid digits are 0 and 1.

binary format: Representation of a decimal value in which each field must be 2 or 4 bytes long. The sign (+ or -) is in the leftmost bit of the field, and the integer value is in the remaining bits of the field. Positive numbers have a 0 in the sign bit and are in true form. Negative numbers have a 1 in the sign bit and are in twos complement form.

binary synchronous communications: A form of communications line control that uses transmission control characters to control the transfer of data over a communications line. Abbreviated BSC. Contrast with *Synchronous Data Link Control.*

browse: In SEU, to look at records in a source member or in a spooled output file by using the record-positioning operations, the scan operation, and the roll keys.

browse member: The member displayed in the lower part of the split-edit display. Records from the browse member can be copied, but no changes can be made to records in the browse member.

BSC: See *binary synchronous communications.*

BSC file: A device file created by the user to support BSC. Contrast with *communications file.*

buffer: A portion of main storage into which data is read or from which it is written.

byte: A group of eight adjacent binary digits that represents one EBCDIC character.

CANCEL: An SNA command used to cancel a partially transmitted RU chain in the network.

card file: A device file created by the user to support a card device.

CF key: See *command function key.*

changed record: An active subfile record to which a put or update operation has been issued with the DDS keyword SFLNXTCHG in effect, or an active subfile record that has been changed by the work station user.

character: Any letter, digit, or other symbol in the data character set that is part of the organization, control, or representation of data.

character field: An area that is reserved for a particular unit of information and that can contain any of the characters in the data character set. Contrast with *numeric field.*

character literal: A symbol, quantity, or constant in a source program that is itself data, instead of a reference to data. Contrast with *numeric literal.*

character set: All the valid characters for a programming language or for a computer system.

character string: (1) A string consisting of any of the 256 EBCDIC characters that are used as a value. (2) (ANSI) In COBOL, a sequence of characters that form a COBOL word, a literal, a PICTURE character-string, or a comment-entry.

CL: See *control language.*

close: A data manipulation function that ends the connection between a file and a program. Contrast with *open.*

collating sequence: The order each character holds in relation to other characters according to the bit structure.

column: A character position horizontally within a print line. The columns are numbered from 1, by 1, starting at the leftmost character position of the print line and extending to the rightmost position of the print line.

command: (1) A statement used to request a function of the system. A command consists of the command name, which identifies the requested function, and parameters. (2) In SNA, any field set in the transmission header (TH), request header (RH), and sometimes portions of a request unit that initiates an action or that begins a protocol.

command function key: At a work station, a keyboard key that is used with the command (CMD) function control key to request preassigned functions. At the system console, a keyboard key, called a CF key, that is used to request preassigned functions.

comment: A word or statement in a program, command, or file that serves as documentation instead of as instructions. A comment is ignored by a compiler.

communications file: A device file created by the user to support LU1 SDLC communications. Contrast with *BSC file.*

communications line: The physical link (such as a wire or a telephone circuit) that connects one or more work stations to a communications control unit, or connects one control unit to another. Contrast with *data link.*

compile: To translate a source program into an executable program (an object).

compile time: The time during which a source program is translated by a compiler into an executable program.

compiler listing: A printout that is produced by compiling a program or creating a file and that optionally includes, for example, a line-by-line source listing, a cross-reference list, diagnostic information, and, for programs, the description of externally described files.

composite key: A key for a file or record format that is composed of more than one key field.

concatenated field: Two or more fields from a physical file record format that have been combined to make one field in a logical file record format.

condition name: For display files, a name used to control the selection of DDS keywords and display locations based on the screen size associated with the display file.

conditioning: (1) In a file, the use of indicators to control when certain functions or operations are to be performed. For example, in a display file, indicators can select fields to be displayed. (2) In an RPG program, the use of indicators to control when certain functions or operations are to be done. For example, in an RPG program indicators can control calculation or output operations.

constant: Data that has an unchanging, predefined value to be used in processing. A constant does not change during the execution of a program, but the contents of a field or variable can. See also *literal.*

constant field: In an externally described display or printer file, an unnamed field that contains actual data that is passed to the display or printer but is unknown to the program passing it.

continuation lines: (1) Additional lines required to continue the coding of a CL command or a DDS keyword and its value. (2) In RPG, additional lines specified on the file description specifications to provide more information about the file being defined.

control language: The set of all commands with which a user requests functions. Abbreviated CL.

control language program: An executable object that is created from source consisting entirely of control language commands.

Control Program Facility: The system support licensed program for System/38. It provides many functions that are fully integrated in the system such as work management, data base data management, job control, message handling, security, programming aids, and service. Abbreviated CPF.

CPF: See *Control Program Facility.*

create: (1) The function used to bring an object into existence in the system. (2) To bring an object into existence in the system.

cross-reference listing: The portion of the compiler listing that contains information on where files, fields, and indicators are defined, referenced, and modified in a program.

currency symbol: A character such as the dollar sign ($) used to identify monetary values.

cursor: A movable spot of light, resembling a bright underscore, that shows where the next character will appear on the work station screen when a key on the keyboard is pressed.

data base: The collection of all data base files stored in the system.

data base file: An object that contains descriptions of how input data is to be presented to a program from internal storage and how output data is to be presented to internal storage from a program. See also *physical file* and *logical file.*

data character set: The 256 EBCDIC characters.

data communications: The transmission of data between systems and/or remote devices over a communications line.

data communications file: In COBOL, a generic term for a communications file or a BSC file. See also *communications file* and *BSC file.*

data description specifications: A description of the user's data base or device files that is entered into the system using a fixed-form syntax. The description is then used to create files. Abbreviated DDS.

data file: Any nonsource file. A data file is created by the specification of FILETYPE(*DATA) on a create file command.

data file utility: The utility of the Interactive Data Base Utilities licensed program that is used to create, maintain, and display records in a data base file. Abbreviated DFU.

data link: The communications lines, modems, control units, work stations, and other communications equipment used for the transmission of data between a receiving station and a transmitting station in a data network. Contrast with *communications line.*

data stream: For BSC, all data transmitted over a data link in a single read or write operation.

data type: An attribute used for defining data as numeric or character.

DDS: See *data description specifications.*

default value: A value assumed when no value has been specified.

definite response: In SNA, a value in the form-of-response-requested field of the request header (RH). The value directs the receiver of the request to return a response unconditionally, whether positive or negative to that request. Contrast with *exception response.*

delay maintenance: A method of maintaining keyed access paths for data base files. This method does not update an access path when the file is closed, but it retains updates in a *delayed* form so that they can be quickly applied at the next open, avoiding a complete rebuild. Contrast with *rebuild maintenance* and *immediate maintenance.*

delete: (1) To remove an object or a unit of data (such as character, a field, or a record). (2) The SEU operation in which existing records can be removed from a source member.

descending key sequence: The arrangement of data in order from the highest value of the key field to the lowest value of the key field. Contrast with *ascending key sequence.*

descending sequence: The arrangement of data in an order from high to low based on the contents of a specific field. Contrast with *ascending sequence.*

DEVD: See *device description.*

device description: An object that contains information describing a particular device that is attached to the system. The system-recognized identifier for the object type is *DEVD. Abbreviated DEVD.

device file: An object that contains a description of how input data is to be presented to a program from an external device and/or how output data is to be presented to the external device from the program. External devices can be work stations, card devices, printers, diskette magazine drives, magazine tape drives, or a communications link.

DFU: See *data file utility.*

digit: Any of the numerals from 0 through 9.

diskette drive: The mechanism used to seek, read, and write data on diskettes. See also *diskette magazine drive.*

diskette file: A device file created by the user to support a diskette device.

diskette magazine drive: A diskette drive that can hold two magazines, each containing 10 diskettes, plus individual diskettes in three separate slots. It is used to transfer information between system internal storage and removable diskettes.

display: A visual presentation of information on a work station screen, usually in a specific format. Display is often used as a shortened version of information display.

display file: A device file created by the user to support a display work station or console.

display screen: An electronic display tube, similar to a TV picture tube, used to display information entered or received at the system console or a work station.

display station: An input/output device containing a display screen and an attached keyboard that lets a user send information to or receive information from the system.

duplicate key value: The occurrence of the same value in a key field or in a composite key in more than one record in a file.

edit: (1) To modify a numeric field to an external format by suppressing zeros and inserting commas, periods, currency symbols, the sign status, or other constant information. (2) The process of using SEU to key in new source records and update existing source records in a source member.

edit code: A letter or number indicating what kind of editing should be done before a field is displayed or printed.

edit description: An object that contains a description of a user-defined edit code. The system-recognized identifier for the object type is *EDTD.

edit word: A user-defined word with a specific format that indicates how editing should be done.

embedded blank: A blank that appears between characters.

end-of-text character: In BSC, the transmission control character that is used to end a logical set of records that began with the start-of-text character. Abbreviated ETX. Contrast with *end-of-transmission-block character.*

end-of-transmission character: In BSC, the transmission control character that is usually used to end transmission with the remote system. Abbreviated EOT.

end-of-transmission-block character: In BSC, the transmission control character used to end a block of records. Abbreviated ETB. Contrast with *end-of-text character.*

ending attribute character: For a display file, the attribute character following the last position in a field.

ENQ: See *enquiry character.*

enquiry character: In BSC, a transmission control character used to indicate control of transmission on a point-to-point channel and to indicate a request for a station to repeat its response.

enter: To press the Enter/Rec Adv key (on a work station keyboard) or the Enter key (on the system console) or a command function key to transfer keyed-in information to the system for processing. See also *key in.*

EOT: See *end-of-transmission character.*

ETB: See *end-of-transmission-block character.*

ETX: See *end-of-text character.*

exception response: In SNA, a value in the form-of-response-requested field of a request header. The value requests the receiver to return a response only if the request is unacceptable as received or cannot be processed; that is, only a negative response can be returned. Contrast with *definite response.*

execution time: The time during which the instructions of a computer program are executed by a processing unit.

external message queue: A message queue that is part of the job message queue and is used to send messages between an interactive job and the work station user. For batch jobs, messages sent to the external message queue only appear in the job log.

externally described data: Data contained in a file for which the fields in the records are described to CPF, by using data description specifications, when the file is created. The field descriptions can be used by the program when the file is processed. Contrast with *program-described data.*

externally described file: A file for which the fields in the records are described to CPF, through data description specifications, when the file is created. The field descriptions can be used by the program when the file is processed. Contrast with *program-described file.*

field: An area that is reserved and used for a particular item of information.

field reference file: A physical file that contains no members and whose record format describes the fields used by a group of files.

field selection: The function of using option indicators on fields in a display file to display different data on different output operations rather than defining a record format for each combination of fields.

file: A generic term for the object type that refers to a data base file, a device file, or a set of related records treated as a unit. The system-recognized identifier for the object type is *FILE.

file description: (1) The information contained in the file that describes the file and its contents. (2) In COBOL, an entry in the File Section of the Data Division that provides information about the identification and physical structure of a file.

first-level message: The initial message that is presented to the user. The initial message contains general information or designates an error. Contrast with *second-level message.*

fixed currency symbol: A currency symbol that appears in a specified position of an edited field. Contrast with *floating currency symbol.*

floating currency symbol: A currency symbol that appears immediately to the left of the leftmost position in an edited field. Contrast with *fixed currency symbol.*

fold: To continue data for a line on the following printed or displayed line. Contrast with *truncate.*

function key: A keyboard key that is used to request a specific system function. See also *command function key.*

get operation: An input operation that obtains a record from an input file and passes it to a program.

help text: Information that is associated with an information display, a menu, or a prompt that explains options or values displayed. Help text is requested by pressing the Help key.

hexadecimal: Pertaining to a numbering system with a base of 16. Valid numbers are the digits 0 through 9 and the characters A through F, where A represents 10 and F represents 15.

hidden field: A field in a display file that is passed from and to the program but is not sent to the display.

high-level language: A programming language that relieves the programmer from the rigors of machine level or assembler level programming; for example, RPG III and COBOL. Abbreviated HLL.

HLL: See *high-level language.*

IDU: See *Interactive Data Base Utilities.*

immediate maintenance: A method of maintaining keyed access paths for data base files. This method updates the access path whenever changes are made to the data in the access path. Contrast with *rebuild maintenance* and *delay maintenance.*

inactive record: A subfile record that has not been added to a subfile by a put operation or that was initialized as inactive by the DDS keywords SFLINZ and SFLRNA.

indicator: (1) A 2-character entry on a specification form that is used to test a field or record or to tell when certain operations are to be performed. (2) An internal switch used by a program to remember when a certain event occurs and what to do when the event occurs.

initialize: To set to a starting position or value.

inline data file: A file described by a //DATA command that is included as part of a job when the job is read from an input device by a reader program.

input: Information (or data) to be processed.

input-capable field: A field in a display file into which data can be entered. An input field is passed from the device to the program when the program reads the record containing that field.

input file: (1) A data base or device file that has been opened with the option to allow records to be read. (2) (ANSI) In COBOL, a file that is opened in the input mode.

input stream: A group of records submitted to the system as batch input that contains CL commands for one or more jobs and/or the data records for one or more inline data files.

inquiry: A request for information from a data file usually made against one record.

integer: (1) Any of the natural numbers, the negatives of these numbers or zero. (2) In COBOL, a numeric data item or literal that does not include any character positions to the right of the decimal point. Where the term integer appears in formats, integer must be an unsigned numeric literal and must be nonzero unless the rules for that format explicitly state otherwise.

interactive: Pertaining to a program or system that alternately accepts input and then responds. An interactive system is conversational; that is, a continuous dialog exists between the user and the system.

Interactive Data Base Utilities: A System/38 licensed program that consists of DFU, SEU, query, and SDA. Abbreviated IDU.

interactive job: A job in which the processing actions are performed in response to input provided by a work station user. During a job, a dialog exists between the user and the system.

interactive processing: Pertaining to a program or procedure that alternately accepts input and then responds to the input. Contrast with *batch processing*.

internal storage: All main and auxiliary storage in the system.

job: A single, identifiable sequence of processing actions that represents a single use of the system. A job is the basic unit by which work is identified on the system.

job date: The date associated with a job. The job date usually defaults to the system date.

job description: An object that contains information defining the attributes of a job. The system-recognized identifier for the object type is *JOBD.

job log: A record of requests submitted to the system by a job, the messages related to the requests, and the actions performed by the system on the job. The job log is maintained by CPF.

job message queue: A message queue that is created for each job. A job message queue is used for receiving requests to be processed (such as commands) and for sending messages that result from processing the requests. A job message queue consists of an external message queue and a set of program message queues. See also *external message queue* and *program message queue*.

job name: The name of a job as identified to the system. For an interactive job, the job name is the name of the work station at which the job was initiated; for a batch job, the job name is specified in the command used to submit the job. Contrast with *qualified job name*.

Julian date: A date format that contains the year in positions 1 and 2, and the day in positions 3 through 5. The day is represented as 1 through 366, right-adjusted, with zeros in the unused high-order positions.

key field: A field in a record whose contents are used to sequence the records of a particular type within a file member.

key in: The action of pressing keys on a keyboard to specify information that is to be processed. See also *enter*.

keyed sequence: The order in which records appear in an access path. The access path is based on the contents of one or more key fields contained in the records.

keyed sequence access path: An access path to a data base file that is ordered on the contents of key fields contained in the individual records. Contrast with *arrival sequence access path*.

keyword: (1) A name that identifies a parameter. Keywords are used in CL commands and in DDS. (2) In RPG, a word whose use is essential to the meaning and structure of a statement in a programming language.

left-adjust: To place an entry in a field or to move the contents of a field so that the leftmost character of the data is in the leftmost position of the field.

level checking: A function that compares the record format level identifiers of a file to be opened with the file description that is part of a compiled program to determine if the file record format has changed since the program was compiled.

library: An object that serves as a directory to other objects. A library is used to group related objects and to find objects by name when they are used. The system-recognized identifier for the object type is *LIB.

library list: An ordered list of library names used to find an object. The library list indicates which libraries are to be searched and the order in which they are to be searched. The system-recognized identifier is *LIBL. *LIBL specifies to the system that a job's current library list is to be used to find the object.

line traffic: The number of transmissions, and the amount of data sent and received on a communications line.

listing: A printout usually containing the input and output of the compilation of a program, the creation (compilation) of an object, or the execution of a program. See also *compiler listing.*

literal: A character string whose value is given by the characters themselves. For example, the numeric literal 7 has the value 7, and the character literal 'characters' has the value CHARACTERS. See also *character literal, constant,* and *numeric literal.*

local work station: A work station that is connected directly to System/38 without need for data transmission facilities. Contrast with *remote work station.*

logical file: A description of how data is to be presented to or received from a program. This type of data base file contains no data, but it provides an ordering and format for one or more physical files. Contrast with *physical file.*

logical file member: A logical grouping of data records in a logical file. See also *member.*

magazine: A container that holds up to 10 diskettes and is inserted into a diskette magazine drive.

MDT: See *modified data tag.*

member: A description of a named subset of records in a physical or logical file. Each member conforms to the characteristics of the file and has its own access path. All I/O requests are directed to a specific member of a data base file.

menu: A display in which a list of options is shown.

message: A communication sent from one person or program to another person or program.

message field: In a display file, an output field that is treated as a message.

message file: An object that contains message descriptions. The system-recognized identifier for the object type is *MSGF.

message identifier: A 7-character code that identifies a predefined message and is used to retrieve its message description from a message file.

message reference key: A key assigned to every message on a message queue. This key is used to remove a message from a message queue, to receive a message, and to reply to a message.

modified data tag: An indicator, associated with each input or output/input field in a displayed record, that is set on when data is keyed into the field. The modified data tag is maintained by the display device and can be used by the program using the file. Abbreviated MDT.

modulus 10 checking/modulus 11 checking: A technique for validity checking that involves the association of digits with data. It is used in entering or updating fields in a data record.

negative response: In SNA, a response indicating that a request did not arrive successfully or was not processed successfully by the receiver. Contrast with *positive response.* See also *exception response.*

numeric character: Any one of the digits 0 through 9.

numeric field: An area that is reserved for a particular unit of information and that can contain only the numeric digits 0 through 9. Contrast with *character field.*

numeric literal: The actual numeric value to be used in processing, instead of the name of a field containing the data. A numeric literal can contain any of the numeric digits 0 through 9, a sign (plus or minus), and a decimal point. Contrast with *character literal.*

object: A named unit that consists of a set of attributes (that describe the object) and, in some cases, data. An object is anything that exists in and occupies space in storage and on which operations can be performed. Some examples of objects are programs, files, and libraries.

object name: The name of an object. Contrast with *qualified object name.*

ODP: See *open data path.*

omit function: A CPF function that determines which records from a physical file are to be omitted from a logical file's access path. Contrast with *select function.*

open: The function that connects a file to a program for processing. Contrast with *close.*

open data path: The path through which all I/O operations for the file are performed. Abbreviated ODP.

operational rights: The authority to use an object and to look at its description.

option indicator: A 1-character field that is passed with an output data record from a program to CPF and that is used to control the output function, such as controlling which fields in the record are displayed.

output: Data transferred from storage to an output device.

output field: A field in a display or printer file in which data can be modified by the program and sent to the display or printer during an output operation.

output file: (1) A data base or device file that has been opened with the option to allow records to be written. (2) In COBOL, a file that is opened in either output mode or extend mode.

output/input field: A field in a display file that is used for both output and input operations.

overflow: The condition that occurs when the last line specified as the overflow line to be printed on a page has been passed.

overflow condition: (1) The condition that occurs when the overflow line on a page has been printed or passed. (2) In COBOL, a condition that occurs when a portion of the result of an operation exceeds the capacity of the intended unit of storage.

overflow handling: The method of advancing from one printer page to the next.

overflow indicator: An indicator that signals when the overflow line on a page has been printed or passed. The indicator can be used to specify which lines are to be printed on the next page.

overflow line: The line specified as the last line to be printed on a page.

overflow page: The new page created when overflow occurs.

overlapping fields: Fields in the same display or printer record that are defined to overlap each other; that is, to occupy the same positions on the display or the page. Option indicators can be used to select which of the overlapping fields is to be displayed or printed.

packed decimal format: Representation of a decimal value in which each byte within a field represents two numeric digits except the rightmost byte, which contains one digit in bits 0 through 3 and the sign in bits 4 through 7. For all other bytes, bits 0 through 3 represent one digit; bits 4 through 7 represent one digit. For example, the decimal value +123 is represented as 0001 0010 0011 1111. Contrast with *zoned decimal format.*

packed field: A field that contains data in the packed decimal format.

packed key: A key in the packed decimal format.

page: (1) A 512-byte block of information that can be transferred between auxiliary storage and main storage. (2) Each group of records in a subfile that are displayed concurrently. (3) One printer form.

parameter: (1) Data passed to or received from another program. (2) In CPF, an argument that identifies an individual value or group of values to be used by a command to tailor a function requested through the command.

physical file: A description of how data is to be presented to or received from a program and how data is actually stored in the data base. A physical file contains one record format and one or more members. Contrast with *logical file.*

physical file member: A subset of the data records in a physical file. See also *member.*

positive response: In SNA, a response indicating that a request arrived and was successfully received and processed. Contrast with *negative response.* See also *definite response.*

printer file: A device file created by the user to support a printer device.

printer/display layout: A coding form on which the programmer can design the format for a printed report or a display.

program: An object that contains a set of instructions that tell a computer where to get input, how to process it, and where to put the results. A program is created as a result of a compilation. The system-recognized identifier for the object type is *PGM.

program-described data: Data contained in a file for which the fields in the records are described in the program that processes the file. Contrast with *externally described data.*

program-described file: A file for which the fields in the records are described only in the program that processes the file. To CPF, the record is viewed as a character string. Contrast with *externally described file.*

program message queue: A message queue used to hold messages that are sent between program invocations of a routing step. The program message queue is part of the job message queue.

prompt: A displayed request for information or user action. The user must respond to allow the program to proceed.

protected field: A field in a display file in which data cannot be keyed, changed, or erased.

put operation: An output operation that writes a record to an output file.

put-get operation: A combination of an output operation (put) followed by an input operation (get) to the same record format.

qualified object name: An object name and the name of the library containing the object. Contrast with *object name.*

query: (1) A utility that is part of the Interactive Data Base Utilities licensed program. (2) A request to extract, from a file, one or more records based upon some combination of data.

queue: A line or list formed by items in the system waiting for service; for example, work to be performed or messages to be displayed.

reader: A program that reads jobs from an input device or a data base file and places them on a job queue.

rebuild maintenance: A method of maintaining keyed access paths for data base files. This method updates the access path only while the file is open, not when the file is closed; the access path is rebuilt when the file is opened. Contrast with *immediate maintenance* and *delay maintenance.*

record: (1) An ordered set of fields that make up a single occurrence of the basic unit of data transferred between a file and a program. (2) In COBOL, a set of one or more related data items that are grouped for processing. Records can be defined for an input/output device or for internal processing.

record format: The definition of how data is structured in the records contained in a file. The definition includes the record name, field names, and field descriptions (such as length and data type). The record formats used in a file are contained in the file's description.

record format level identifier: An identifier placed on a record format that uniquely identifies the record description. See also *level checking.*

relational operator: (1) In CL, an operator that can be used in an arithmetic, character, or logical relation to indicate the comparison to be performed between the terms in the relation. The relational operators are *EQ or = (equal to), *GT or > (greater than), *LT or < (less than), *GE or >= (greater than or equal to), *LE or <= (less than or equal to), *NE or ¬= (not equal to), *NG or ¬> (not greater than), *NL or ¬< (not less than). (2) In COBOL, a reserved word, a relation character, a group of consecutive reserved words, or a group of consecutive reserved words and relation characters used to construct a relation condition.

remote work station: A work station whose connection to the processing system uses modems and common carrier or private data transmission facilities. Contrast with *local work station.*

request: (1) A CL command, the selection of an option on a menu, or the entering of data that instructs the system to perform a function. A CL command can be entered interactively or in a batch job. A request is identified as RQS on the job log. (2) In SNA, a request unit that asks for a particular action or protocol; or a message unit that acknowledges receipt of a request. A request consists of a request unit, a segment of a request unit, or both.

response: In SNA, a message unit that acknowledges receipt of a request; a response consists of a response header, a response unit, or both.

response indicator: A 1-character field passed with an input record from CPF to a program to provide information about the data record or actions taken by the work station user.

reverse-interrupt character: In BSC, a transmission control character that is sent as a request by the receiving station to the sending station to stop transmitting in order to receive a message. Abbreviated RVI.

right-adjust: To place an entry in a field or to move the contents of a field so that the rightmost character of the data is in the rightmost position of the field.

routing data: A character string that CPF compares with character strings in the subsystem description routing entries to select the routing entry that is to be used to initiate a routing step. Routing data can be provided by a work station user, specified in a command, or provided through the job description for the job.

routing entry: An entry in a subsystem description that specifies the program to be invoked to control a routing step that executes in the subsystem.

routing step: The processing performed as a result of invoking a program specified in a routing entry.

RSP: See *response.*

RVI: See *reverse-interrupt character.*

screen design aid: The utility of the Interactive Data Base Utilities licensed program that is used to interactively design, create, and maintain display record formats and menus. Abbreviated SDA.

SDA: See *screen design aid.*

second-level message: A message that provides additional information to that already provided in a first-level message. See also *second-level message display.*

second-level message display: A display containing the second-level message text (if any) and additional message information. This display is obtained by pressing the Help key while a first-level message is displayed.

security: The control of access to, or use of, data or functions.

select function: A CPF function that determines which records from a physical file are to be selected for a logical file's access path. Contrast with *omit function.*

select/omit field: A field in a logical file record format whose value is compared with a constant, the contents of another field, a range of values, or a list of values to determine if a record is to be omitted from the access path of the logical file or selected for use by the logical file. See also *omit function* and *select function.*

self-check digit: The rightmost digit of a self-check field.

self-check field: A field such as an account number, consisting of a base number and a self-check digit. For data entry applications, the operator-entered self-check digit is compared with the self-check digit computed by the system.

sequence number: The number of a record that identifies the record within the source member.

sequential file: A file in which records are processed in the order that they are stored in the file.

SEU: See *source entry utility.*

severity code: A code that indicates how important a message is. The higher the code, the more serious the condition is.

shared access path: An access path used by more than one file to provide access to data common to the files.

shared file: A file whose open data path can be shared between two or more programs executing in the same routing step.

shared record format: A record format that is used in more than one externally described file.

SIGNAL: An SNA command used to request a break in data flow.

skip: To cause the printer to move the paper to a specified line before or after it prints a line.

SNA: See *Systems Network Architecture.*

source entry utility: The utility of the Interactive Data Base Utilities licensed program that is used to create and change source members. Abbreviated SEU.

source file: A file created by the specification of FILETYPE(*SRC). A source file can contain source statements for such items as high-level language programs and data description specifications.

source listing: A portion of a compiler listing that contains source statements and diagnostics. See also *compiler listing.*

source member: A member of a data base source file that contains source statements such as RPG, COBOL, or DDS specifications. See also *member.*

source statement: A statement written in symbols of a programming language. For example, RPG, COBOL, or DDS specifications are source statements.

space: To cause the printer to move the paper a specified number of lines before or after it prints a line.

special character: (1) A character other than a digit, a letter, or #, $, or @. For example, *, +, and % are special characters. (2) In COBOL, a character that is neither numeric nor alphabetic. Special characters in COBOL include: + - * / = $, . ") (; < >

spooled file: A generic term for three types of files: a device file that provides access to an inline data file or that creates a spooled output file, an inline data file, or a spooled output file.

spooled input file: See *inline data file.*

spooled output file: A device file that causes output data to be saved for later processing by a writer.

start-of-text character: In BSC, a transmission control character that is used to begin a logical set of records that will be ended by the end-of-text character. Abbreviated STX.

STX: See *start-of-text character.*

subfile: A group of records of the same record format that can be displayed concurrently at a work station. The system sends the entire group of records to the work station in a single operation and receives the group in another operation.

subfile control record format: One of two record formats required to define a subfile in DDS. The subfile control record format describes the size of the subfile and the size of the subfile page, and is used by the program to write the subfile to and read the subfile from the display. See also *subfile record format.*

subfile record format: One of two record formats required to define a subfile in DDS. The subfile record format defines the fields in a subfile record and is used by the program to perform input, output, and update operations to the subfile.

Synchronous Data Link Control: A discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-level Data Link Control (HDLC) of the International Standards Organization (ISO), for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. Abbreviated SDLC.

syntax checking: A function of the command analyzer, a compiler, or SEU that checks single statements for violations of the rules governing the structure of the statement.

system console: The keyboard and display screen on the system unit that serve as a work station for communicating with and controlling the system. See also *work station.*

system date: The date established for the system when it is started.

system operator: The person who operates the system and looks after the peripheral equipment necessary to initiate computer runs or finalize the computer output in the form of completed reports and documents.

system value: A value that contains control information for the operation of certain parts of the system. A user can change the system default value to tailor the system to his working environment. System date and library list are examples of system values.

Systems Network Architecture: The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of Systems Network Architecture networks. Abbreviated SNA. **Note:** The layered structure of SNA allows the ultimate origins and destinations of information (that is, the end users) to be independent of, and unaffected by, the specific SNA network services and facilities used for information exchange.

tape file: A device file created by the user to support a tape device.

truncate: To drop data that cannot be printed or displayed in the line width specified or available. Contrast with *fold*.

update operation: An I/O operation that modifies the information in a file.

user-defined edit code: A number (5 through 9) indicating that editing should be done on a numeric output field according to a pattern predefined to CPF. User-defined edit codes can take the place of edit words, so that repetitive coding of the same edit word is not necessary.

user profile: An object that contains a description of a particular user or group of users. A user profile contains a list of authorizations to objects and functions. The system-recognized identifier for the object type is *USRPRF.

validity checker: A user-written program that tests commands for errors in the parameter values. Validity checking is done in addition to the checking done by the command analyzer.

validity checking: Operations performed against a field value to ensure that the field contains appropriate data. Checking can be done on a single field (for example, the field must be plus) or on multiple fields (for example, if FLDA contains a 1, FLDB can contain only a 2 or 3).

variable: A named modifiable value. The value can be accessed or modified by referring to the name of the variable.

vertically displayed records: Subfile records that are grouped in a display so that more than one record of the same record format is displayed concurrently. Each record begins in the first position of a line and occupies one or more adjoining lines.

work station: A device that lets a person transmit information to or receive information from a computer as needed to perform his job.

work station user: A person who uses a work station to communicate with System/38.

zoned decimal format: Representation of a decimal value by 1 byte per digit. Bits 0 through 3 of the rightmost byte represent the sign; bits 0 through 3 of all other bytes represent the zone portion; bits 4 through 7 of all bytes represent the numeric portion. For example, in zoned decimal format, the decimal value of +123 is represented as 1111 0001 1111 0010 1111 0011. Contrast with *packed decimal format*.

zoned field: A field that contains data in the zoned decimal format.

Index

READER'S COMMENT FORM

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your nearest IBM branch office.

☐ If your comment does not need a reply (for example, pointing out a typing error) check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

☐ If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s): Comment(s):

Please contact your nearest IBM branch office to request additional publications.

Name _____

Company or Organization _____

Address _____

_____

City State Zip Code

IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

No postage necessary if mailed in the U.S.A.

Cut Along Line

Fold and tape | Please do not staple | Fold and tape

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N. Y.

POSTAGE WILL BE PAID BY ADDRESSEE

IBM CORPORATION
General Systems Division
Development Laboratory
Publications, Dept. 245
Rochester, Minnesota 55901

Fold and tape | Please do not staple | Fold and tape

IBM

International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30055
(U.S.A. only)

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

IBM System/38 Control Program Facility Reference Manual—Data Description Specifications Printed in U.S.A. SC21-7806-4

International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30055
(U.S.A. only)

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

IBM System/38 Control Program Facility Reference Manual—Data Description Specifications Printed in U.S.A. SC21-7806-4