



**DATA CATALOGUE 2
REFERENCE MANUAL**

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

PREFACE

Data Catalogue 2 operates under control of the following operating systems:

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

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

Data Catalogue 2 is a CDC modification of the product designed and developed by Synergetics Corporation.

This manual describes the standard facilities available to Data Catalogue users. It is intended to be the primary reference for users of the product. It is assumed that the user of this manual is familiar with the system under which Data Catalogue 2 is operating. Related material is contained in the publications listed below.

<u>Publication</u>	<u>Publication Number</u>
CYBER Record Manager Advanced Access Methods Version 2 Reference Manual	60499300
Data Catalogue 2 Data Administrator's Reference Manual	60483200
NOS Version 1 Reference Manual, Volume 1 of 2	60435400
NOS Version 1 Reference Manual, Volume 2 of 2	60445400
NOS/BE Version 1 Reference Manual	60493800

CDC manuals can be ordered from Control Data Corporation, Literature and Distribution Services, 308 North Dale Street, St. Paul, Minnesota 55103.

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

CONTENTS

NOTATIONS USED IN THIS MANUAL	xi		
1. INTRODUCTION TO DATA CATALOGUE	1-1	TOTAL Dataset Entity	3-17
Data Catalogue Terminology	1-1	RESPONSIBILITY Category	3-18
Data Entities	1-1	NAMES Category	3-18
Procedure Entities	1-2	ATTRIBUTES Category	3-19
User Entity	1-2	UNIT Category	3-20
Data Catalogue Structures	1-2	ENVIRONMENT Category	3-21
Data Catalogue Functions	1-2	STRUCTURE Category	3-22
Update Function	1-3	TOTAL Data Base Entity	3-23
Query Function	1-3	RESPONSIBILITY Category	3-23
Report Function	1-3	NAMES Category	3-24
Generate Function	1-4	IOAREA Category	3-24
Convert Function	1-5	STRUCTURE Category	3-24
Utility Function	1-5	Procedure Entities	3-25
2. DATA CATALOGUE CONVENTIONS	2-1	Form Entity	3-25
Catalogue Entries	2-1	RESPONSIBILITY Category	3-25
Catalogue Names	2-1	NAMES Category	3-26
System Names	2-1	ATTRIBUTES Category	3-26
Report Names	2-1	SUPPLIERS Category	3-26
Category Headers	2-1	FLOW Category	3-28
Category Line Numbers	2-1	RELATIONAL Category	3-29
Continuation Lines	2-1	Report (Screen) Entity	3-29
Comment Lines	2-2	RESPONSIBILITY Category	3-29
Partial Usage	2-2	NAMES Category	3-30
Aliases and Versions	2-2	ATTRIBUTES Category	3-31
Function Requests	2-2	FLOW Category	3-31
Request Statements	2-2	RELATIONAL Category	3-32
Comments	2-2	External Resource Entity	3-33
Statement Continuation	2-2	RESPONSIBILITY Category	3-33
Request Forms	2-3	NAMES Category	3-34
Error Messages	2-3	LOCATION Category	3-34
3. DATA CATALOGUE ENTITIES	3-1	RELATIONAL Category	3-34
Common Categories	3-1	Program (Module) Entity	3-35
CONTROL Category	3-2	RESPONSIBILITY Category	3-35
CLASSIFICATION Category	3-2	NAMES Category	3-36
DESCRIPTION Category	3-2	ATTRIBUTES Category	3-36
OTHER Category	3-2	STORAGE Category	3-37
Data Entities	3-3	OPERATING Category	3-37
Element Entity	3-3	SCHEDULING Category	3-38
ORIGIN Category	3-3	PASSING Category	3-39
VALUES Category	3-4	RELATIONAL Category	3-39
NAMES Category	3-4	Manual Task Entity	3-40
ATTRIBUTES Category	3-5	RESPONSIBILITY Category	3-41
ALIAS Category	3-6	NAMES Category	3-41
Group Entity	3-7	SCHEDULING Category	3-42
NAMES Category	3-7	RELATIONAL Category	3-43
ATTRIBUTES Category	3-8	System Entity	3-43
STRUCTURE Category	3-9	RESPONSIBILITY Category	3-43
Record Entity	3-10	NAMES Category	3-44
NAMES Category	3-10	RELATIONAL Category	3-45
ATTRIBUTES Category	3-10	User Entity	3-45
STRUCTURE Category	3-11	PERSONNEL Category	3-45
TOTAL STRUCTURE Category	3-12	RELATIONAL Category	3-46
File Entity	3-13	4. UPDATE FUNCTION	4-1
RESPONSIBILITY Category	3-13	Data Catalogue File Maintenance	4-1
NAMES Category	3-14	Update Request Structure	4-1
ATTRIBUTES Category	3-14	\$UPDATE Statement	4-1
UNIT Category	3-16	OPTIONS Statement	4-2
STRUCTURE Category	3-17	ADD Statement	4-2
		CHG Statement	4-2
		DEL Statement	4-3
		Category Header Statement	4-3
		Category Line Number Statement	4-4
		Update Error Processing	4-5
		Auditing and Logging Transactions	4-5
		Backup and Restore of System Files	4-5
		Printed Statistics	4-5

5. QUERY FUNCTION	5-1	8. CONVERT FUNCTION	8-1
Query Function Capabilities	5-1	Convert Applications	8-1
Information Retrieval	5-1	Create Entries	8-1
Data Administration	5-1	Data Usage in Modules	8-1
Query Request Structure	5-2	Analysis of Modules	8-1
\$QUERY Statement	5-2	Checklisting Data	8-1
TITLE Statement	5-2	Considerations for Conversion	8-1
COUNT Statement	5-3	Thoroughness of Definitions	8-2
COUNT Statement Phrases	5-3	Commonality of Names	8-2
COUNT Statement Response	5-3	Number of Modules to be Scanned	8-2
LIST Statement	5-3	Coding Techniques	8-2
LIST Statement Phrases	5-4	Convert Request Structure	8-2
LIST Statement Response	5-4	\$CONVERT Statement	8-3
SHOW Statement	5-4	MODULE Statement	8-3
SHOW Statement Phrases	5-4	SELECT Statement	8-4
SHOW Statement Response	5-5	RENAME Statement	8-4
Search Clauses	5-6	CREATE Statement	8-5
WITH Clause	5-6		
HAVING Clause	5-7	9. UTILITY FUNCTION	9-1
USED-BY Clause	5-8	System Initialization	9-1
WHICH-USE Clause	5-9	Installation Identification	9-1
Logical Connectors	5-9	Optional Features	9-1
Range Retrieval	5-10	Options Maintenance	9-1
Entity Type Range	5-10	Options Display	9-1
Catalogue Name Range	5-10	Option Changes	9-1
		Special Facilities	9-1
6. REPORT FUNCTION	6-1	Definition Transfer	9-2
Report Request Structure	6-1	Backup and Restore	9-2
\$REPORT Statement	6-1	System Statistics	9-2
TITLE Statement	6-2	Line Renumbering	9-2
SELECT Statement	6-2	Renaming Entries	9-2
OUTPUT Statement	6-4		
OPTIONS Statement	6-4	10. OPERATING REQUIREMENTS	10-1
Documentation Reports	6-4	Common Considerations	10-1
Catalogue Report	6-4	Executable Programs	10-1
Usage Report	6-5	Pseudo-Switch Settings	10-1
Hierarchy Report	6-7	Common Files	10-1
Analysis Reports	6-8	Functional Considerations	10-2
Index Report	6-8	Utility Function	10-2
Relational Report	6-9	Update Function	10-2
Name Analysis Report	6-10	Required Files	10-2
		Operational Considerations	10-2
7. GENERATE FUNCTION	7-1	Output	10-3
Generate Capabilities	7-1	Query Function	10-3
Generate Request Structure	7-1	Required Files	10-3
\$GENERATE Statement	7-1	Operational Considerations	10-3
OPTIONS Statement	7-2	Output	10-3
SELECT Statement	7-3	Report Function	10-4
Considerations for Generation	7-4	Required Files	10-4
COBOL Considerations	7-4	Operational Considerations	10-6
Basis for Generation	7-4	Output	10-6
Data Name Selection	7-5	Generate Function	10-7
PICTURE Clause	7-5	Required Files	10-7
Filler Items	7-5	Operational Considerations	10-7
Other Entity Fields	7-5	Output	10-8
Definition Errors	7-6	Convert Function	10-8
Generate Request Output	7-6	Required Files	10-8
PL/I Considerations	7-6	Operational Considerations	10-8
Basis for Generation	7-6	Output	10-9
Identifier Selection	7-6		
Data Attributes	7-6	11. EXAMPLES	11-1
Filler Entries	7-6	Update Function	11-1
Other Entity Fields	7-6	Query Function	11-1
Compiler Editing	7-6		
TOTAL DBDL Considerations	7-6		

Report Function	11-1	Hierarchy Report	11-2
Catalogue Report	11-1	Usage Report	11-2
Index Report	11-2	Function Reports	11-2
Relational Report	11-2	Generate Function	11-2
Name Analysis Report	11-2	Convert Function	11-2

APPENDIXES

A Standard Character Sets	A-1	D Summary of Entity Definitions	D-1
B Error Processing and Diagnostics	B-1	E Request Forms	E-1
C Glossary	C-1		

INDEX

FIGURES

1-1 Sample System Structure	1-3	7-4 SELECT Statement Format, Generate Function	7-3
1-2 TOTAL Data Base Structure	1-4	8-1 Conversion with Multiple Record Layouts	8-2
1-3 Typical Group Structure	1-4	8-2 Convert Request Deck Setup, COBOL Conversion	8-3
1-4 Typical File Structure	1-5	8-3 Convert Request Deck Setup, TOTAL DBDL Conversion	8-3
4-1 Update Request Deck Setup	4-1	8-4 \$CONVERT Statement Format	8-3
4-2 OPTIONS Statement Format, Update Function	4-2	8-5 MODULE Statement Format	8-3
4-3 ADD Statement Format	4-2	8-6 SELECT Statement Format, Convert Function	8-4
4-4 CHG Statement Format	4-3	8-7 RENAME Statement Format	8-4
4-5 DEL Statement Format	4-3	8-8 CREATE Statement Format	8-5
4-6 Category Header Statement Format	4-4	10-1 Update Function File Usage	10-2
4-7 Category Line Number Statement Format	4-4	10-2 Update Function Control Statements	10-3
5-1 Query Request Deck Setup	5-2	10-3 Query Function File Usage	10-4
5-2 COUNT Statement Format	5-3	10-4 Query Function Control Statements	10-4
5-3 LIST Statement Format	5-4	10-5 Report Function File Usage	10-5
5-4 SHOW Statement Format	5-5	10-6 Report Function Control Statements	10-6
5-5 SHOW Statement Response Format	5-5	10-7 Generate Function File Usage	10-7
5-6 WITH Clause Format	5-6	10-8 Generate Function Control Statements	10-7
5-7 HAVING Clause Format	5-7	10-9 Convert Function Control Statements	10-8
5-8 USED-BY Clause Format	5-8	10-10 Convert Function File Usage	10-9
5-9 WHICH-USE Clause Format	5-9	11-1 Update Request Example	11-3
5-10 Logical Relationship Format	5-9	11-2 Update Audit Report	11-4
5-11 Entity Type Range Format	5-10	11-3 Query Request Example	11-5
5-12 Catalogue Name Range Format	5-10	11-4 Query Response Report	11-6
6-1 Report Request Deck Setup	6-1	11-5 Report Request Example	11-9
6-2 \$REPORT Statement Format	6-2	11-6 Report Request Audit	11-10
6-3 SELECT Statement Format, Report Function	6-3	11-7 Extract Control Report	11-11
6-4 OUTPUT Statement Format	6-4	11-8 Catalogue Report	11-12
6-5 OPTIONS Statement Format, Catalogue Report	6-4	11-9 Index Report	11-14
6-6 OPTIONS Statement Format, Usage Report	6-6	11-10 Relational Report	11-15
6-7 OPTIONS Statement Format, Hierarchy Report	6-7	11-11 Name Analysis Report	11-16
6-8 OPTIONS Statement Format, Index Report	6-7	11-12 Hierarchy Report	11-17
6-9 OPTIONS Statement Format, Relational Report	6-8	11-13 Usage Report	11-20
6-10 OPTIONS Statement Format, Name Analysis Report	6-9	11-14 Request Control Report, Run Totals	11-22
7-1 Generate Request Deck Setup	6-10	11-15 Index Example	11-22
7-2 \$GENERATE Statement Format	7-1	11-16 Generate Request Example	11-23
7-3 OPTIONS Statement Format, Generate Function	7-2	11-17 Generation Report	11-24
	7-2	11-18 Convert Request Example	11-25
		11-19 Conversion Reports	11-26

TABLES

3-1 Applicable Categories for Entities	3-1	7-1 Summary of Options and Defaults	7-4
3-2 Valid Entity Relationships	3-29	9-1 Initialization Options	9-1
5-1 Entity Types and Categories	5-6	10-1 Catalogue File Characteristics	10-1
5-2 Hierarchical Position of Entity Types	5-9		

NOTATIONS USED IN THIS MANUAL

UPPERCASE

Words are Data Catalogue reserved words. They must be spelled correctly including any hyphens.

items are stacked vertically within brackets, only one of the stacked items can be used.

UPPERCASE

Underlined uppercase words are required when the format in which they appear is used. When a portion of a word is underlined, either the underlined portion or the entire word can be used.

{ }

Braces

Portion of a format in which one, and only one, of the vertically stacked items must be used. Braces are also used to enclose the portion of a required entry that can be repeated.

[]

Brackets

Optional portion of a format. All of the format within the brackets can be omitted or included at user option. If

...

Ellipses

Repetition indicator. Portion of the format enclosed in the immediately preceding braces or brackets can be repeated at user option.

Data Catalogue is a computerized tool for documenting the information processing function. It produces a Catalogue that is a dictionary of data, procedures, and users. The Catalogue is a central repository where information processing documentation can be maintained, accessed, retrieved, and distributed at computer speeds.

The data administration function recognizes that data itself is as much a valuable and manageable commodity as computer programs, computer hardware, and personnel. Data is continually expanding in scope; new data items and new records are continually being added. As the business environment changes, most companies are quick to add new data but slow to remove obsolete data. The quantity of data has increased, is still increasing, and will continue to increase. It is difficult to eliminate data in a large information processing environment, because it is not known whether that data is used in some other application.

Another problem that develops in large data processing environments is duplication and redundancy at the program level as well as at the data item level. In some environments there are thousands of files and thousands of programs accessing the files. The task of managing, organizing, and accessing the various types of information within this environment is compounded by the quantity of information to be described. Each type of entity (such as file, data element, or program) has unique characteristics to be described and stored.

Data Catalogue provides an answer to these problems by computerizing the descriptions and definitions of the data processing environment with a flexible but structured methodology. The structure of Data Catalogue allows information to be stored for each of the various entity types from the system level down to the data element level.

With Data Catalogue, hierarchies can be established through relationships between the various entries in the Catalogue. Following a hierarchy upward or downward provides the data administrator with various types of information. The data that is available and how it relates to other data can be an effective tool in assessing the cost of making changes within the information processing function. The upward hierarchy of a data element, for example, shows the uses of the element in groups, records, files, and programs.

Knowledge is the fundamental management tool. Knowledge gained through Data Catalogue allows the data administrator to make efficient use of the resources in the data processing environment. An analyst writing a new application program can be shown that certain data required by the application already exists in the system. The effect of making a change to a data element can be determined by studying the relationships of other entries to the data element.

A completely mechanized documentation system can be developed by defining and describing all of the data, procedures, and users in the data processing environment. All of the components of a system can be fully described and all of the responsibility relationships can be defined

and stored in the Catalogue. In the process of defining entities to Data Catalogue, the documentation process can be standardized.

Data Catalogue can be used as a scratch pad for tentative definitions. Initial design data structures can be described to Data Catalogue. As the application develops, the tentative definitions can be updated and refined. By the time the design is complete, the data definitions have already been documented and can become a part of the permanent Catalogue.

DATA CATALOGUE TERMINOLOGY

A Data Catalogue entry comprises all the information that relates to a unique type of entity. Entity types are grouped according to the type of information that is used to describe them.

DATA ENTITIES

Data entities can be conveniently represented in hierarchies. Entity types in this group are as follows:

Element

An element is the lowest level data entity in the hierarchy. It can be used in many different structures, but the element does not change in its essence. The format of an element can change depending on whether it is used in a transaction or stored in a file; however, its meaning stays the same wherever it is used. An element cannot be subdivided into other elements.

In the TOTAL data base environment, a Data Catalogue element is a data item. The Data Catalogue element entity is used to describe a TOTAL data item.

Group

A group entity contains two or more subcomponents. These components can be elements, other groups, or a combination of elements and groups. A group is a logical collection of data that can be addressed as a single unit. The group entry SHIP-DATE, for example, consists of the element entries SHIP-MONTH, SHIP-DAY, and SHIP-YEAR.

In the TOTAL data base environment, a Data Catalogue group is an element. The Data Catalogue group entity is used to describe a TOTAL element.

Record

A record is a logically associated collection of elements, groups, or combination of elements and groups. A record entity specifies the elements and groups of which it is comprised. A

record is retrieved or stored in a conventional file or a TOTAL dataset.

File

A file is a collection of one or more records. It can be accessed and stored physically. A single file can contain many different types of records and many different records of each type. A file entity describes the characteristics of the file.

Dataset

The TOTAL dataset is essentially a file of records. The dataset is a part of the data base and can be related to other datasets within the data base. A dataset entity describes the characteristics of a TOTAL dataset.

Data Base

The TOTAL data base is a logical aggregate of data. The data base can be seen as a collection of datasets that participate in a defined relationship. A data base entity describes the characteristics of a TOTAL data base.

PROCEDURE ENTITIES

A procedure entity can have relationships with other procedure entities; it can also interact with data or responsibilities. Procedure entities are as follows:

Module

A module is a computer process that performs one or more discrete tasks. A module can be free-standing or part of a program.

Program

The term program is almost synonymous with module. A program can contain modules, call modules, or consist of one module itself.

System

A system is a self-contained collection of one or more computer programs and ancillary manual tasks that perform a given function completely.

Form

A form is a document containing printed captions, data, and blank spaces that need to be filled in with data. A form can be a source for the entry of data, or it can be partly the output of one function and partly the source of another.

Report

A report is a formatted presentation of information. It can be printed or displayed. A screen could also be referred to as a report.

External Resource

An external resource is one that is not accessible by computer. Relevant resources could be flowcharts, procedural instructions, or other externally stored documents.

Manual Task

A manual task is one that does not involve any computer processing. It might require access to computer-prepared reports, but it is executed by a person. A manual task could be balancing the transaction dollar total or reconciling error report data.

USER ENTITY

A user entity describes the responsibility of a management unit for a given activity or resource. Only one entity type falls into this group:

User

A user entity specifies the department, section, or functional group responsible for a given activity or resource. A management unit can be assigned the responsibility for a system, a program, a manual task, a file, or an external resource.

DATA CATALOGUE STRUCTURES

Data Catalogue recognizes structured relationships, which are an integral part of the information processing function. Looking at information processing from the top down, hierarchies and relationships become evident.

Figure 1-1 is an example of a system showing some hierarchies and relationships. The applicable Data Catalogue entity types are noted in the illustration. Typical systems contain many components with many explicitly or implicitly defined cross relationships between the data resources, the people who carry out the required actions, and the procedures that act upon the resources.

Figure 1-2 illustrates the hierarchy of entities in a TOTAL data base environment. Data Catalogue terminology is used in the illustration. TOTAL terminology, which is in parentheses, is included where applicable. The group entity can contain other groups at any level of nesting desired. This enables documenting TOTAL elements and subelements as well as data items.

An element is the lowest level in a data structure; it cannot be subdivided. A group is comprised of elements and/or subgroups. In figure 1-1, the group ORDER-DATE is a simple group structure consisting of two elements. Figure 1-3 shows a group (GROUP-A) that consists of two elements (ELEMENT-A and ELEMENT-D) and another group (GROUP-B).

The structure of a typical file is shown in figure 1-4. The file known as FILE-A is composed of two record types (RECORD-A and RECORD-B). Both record types are in turn made up of elements and a group. The group known as GROUP-A is defined only once; RECORD-A and RECORD-B need only reference GROUP-A.

DATA CATALOGUE FUNCTIONS

Six functions are provided to perform the operations necessary to create and maintain the Catalogue as well as to access information stored in the Catalogue. The functions are individually described in section 4 through section 9. The operational requirements for the functions are presented in section 10. The following paragraphs provide a brief overview of the facilities available through the Data Catalogue functions.

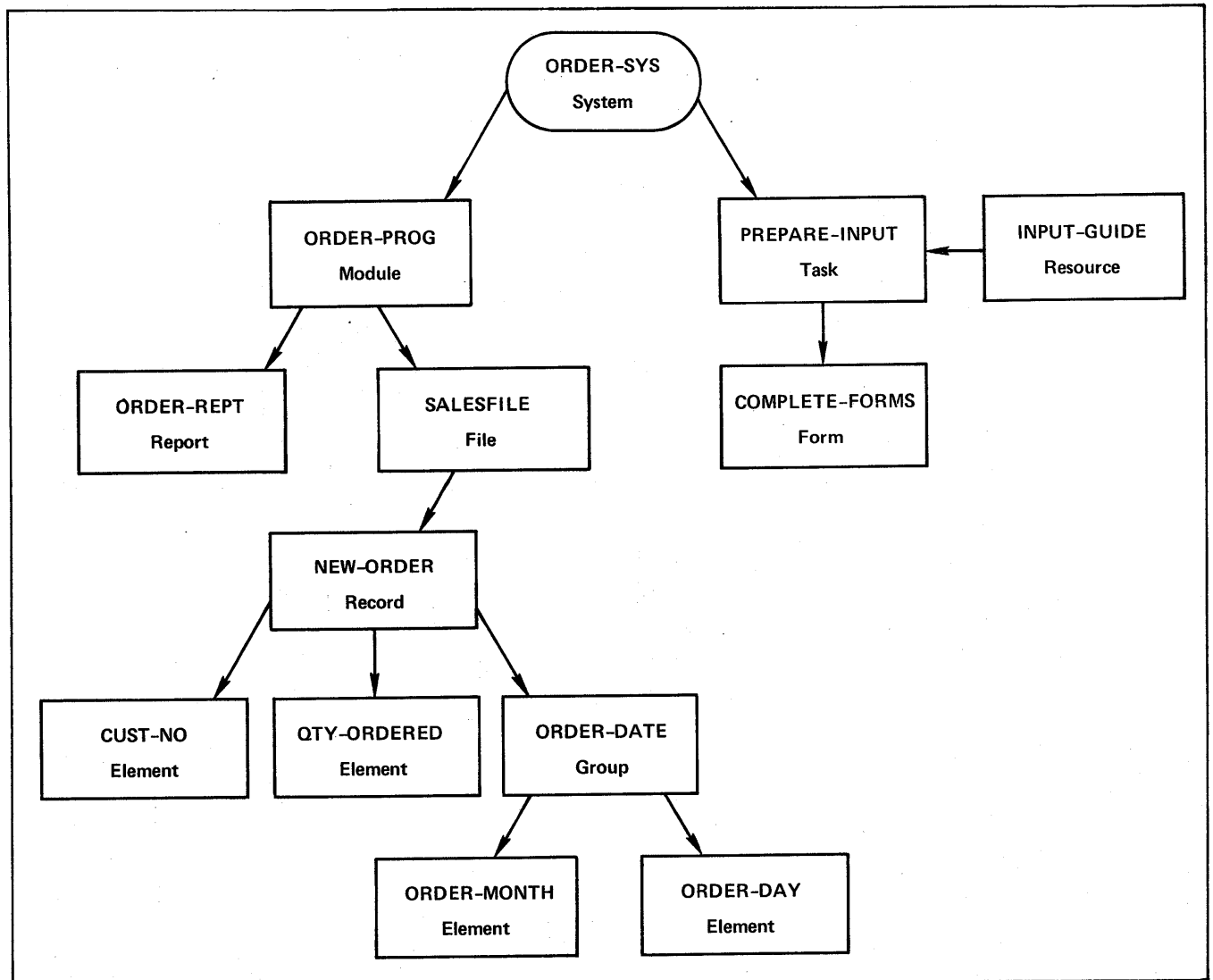


Figure 1-1. Sample System Structure

UPDATE FUNCTION

The Update function is provided to maintain the Catalogue. New entries are created and added to the Catalogue through this function. Existing entries can be changed by adding field descriptions, changing existing field descriptions, and deleting field descriptions. Entire entries can also be deleted from the Catalogue through the Update function.

Output from the Update function includes the updated Catalogue files and an audit report. The audit report is a log of transactions input to the Update function. Error messages are output to the audit report for invalid transactions. The last page of the audit report is a statistical listing of entry and transaction counts.

QUERY FUNCTION

Various types of information about the Catalogue contents can be obtained through the Query function. Query language is used to interrogate the Catalogue. Entries can be counted, listed by Catalogue name, or listed in

detail. Questions can be general (list all elements) or specific (list elements with a specified length).

Output from the Query function is in the form of a response to each question. The amount of output in response to a question depends on the type of information requested. A request for a count of entries produces a one-line response. In addition to the one-line count response, a request for Catalogue names of entries produces one line of output for each qualified entry. The response to a request for a detailed listing of entries includes the one-line count response, the list of Catalogue names, and a detailed description of each qualified entry.

REPORT FUNCTION

The Report function provides the means to produce several different types of printed reports from information stored in the Catalogue. Three reports are used to document the Catalogue. The Catalogue Report shows all the information stored for each selected entry. The Usage Report presents detailed information for an entry and for all higher-level entries that use it. The

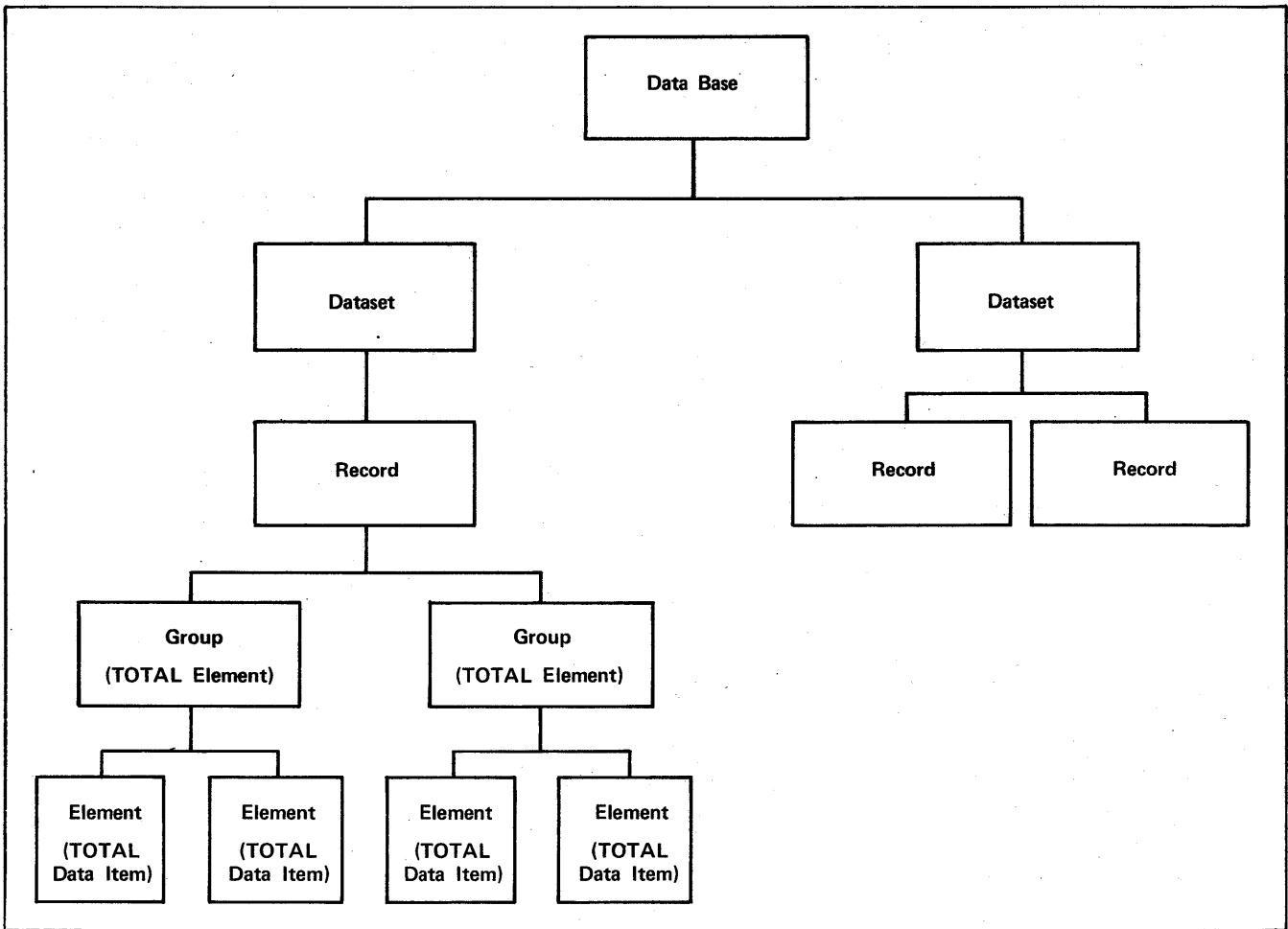


Figure 1-2. TOTAL Data Base Structure

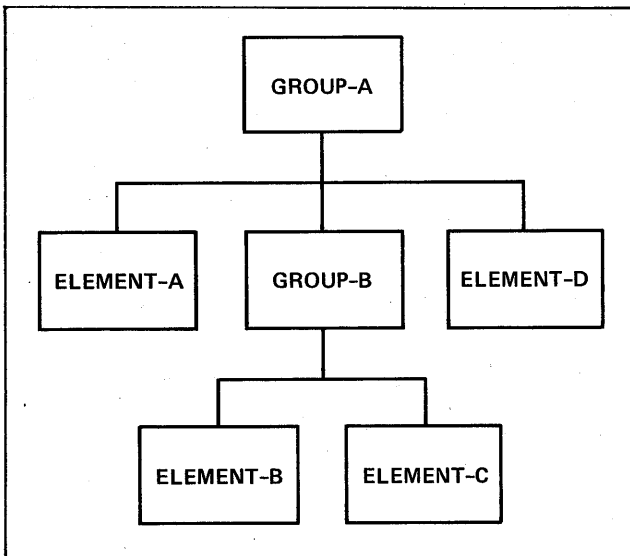


Figure 1-3. Typical Group Structure

Hierarchy Report presents detailed information for an entry and for all lower-level entries that are used by it.

Three reports are used for analysis of the information stored in the Catalogue. The Index Report produces an index for any standard field used to describe entries. The Relational Report provides a cross-reference of entries to standard Catalogue fields. The Name Analysis Report lists in a group compound names that have a common component.

The reports output by the Report function can be printed in standard computer-width format or in 8-1/2 by 11 format for finished documentation. A run time option specifies the output format.

GENERATE FUNCTION

Source statement data definitions are created from information stored in the Catalogue through the Generate function. Definitions can be generated for use in COBOL and PL/I programs. Data Base Definition Language (DBDL) statements can also be generated for the TOTAL user.

The Generate function provides two types of output. A listing of the generated statements can be scanned to ensure that all definitions are complete. The generated statements are output to a sequential file for subsequent use. A run time option allows the user to select either or both types of output.

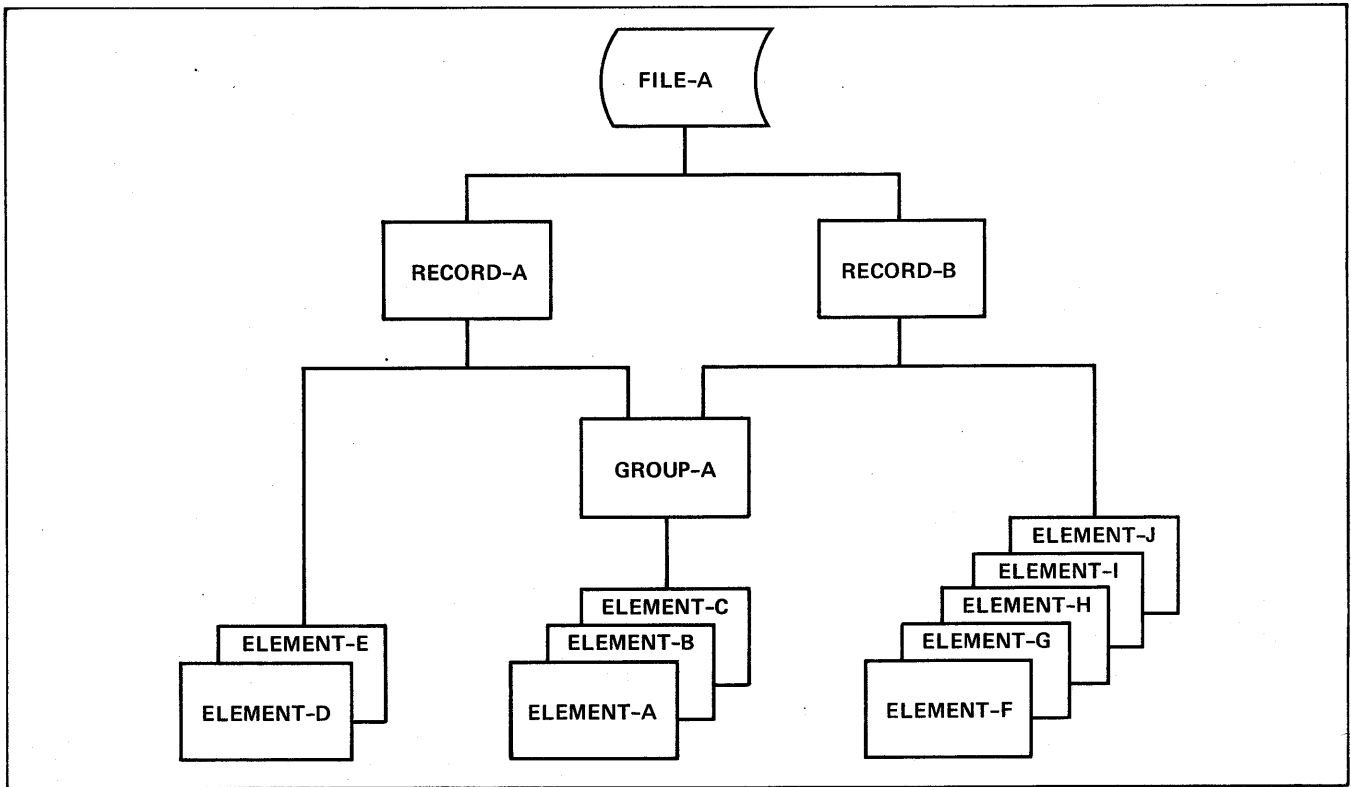


Figure 1-4. Typical File Structure

CONVERT FUNCTION

The Convert function is used to create Data Catalogue definitions for existing COBOL programs and TOTAL DBDLs. The existing program or DBDL is scanned by the function, and input transactions for creating new entries are generated.

Output from the Convert function is a sequential file of transactions that can be input to the Update function. An audit trail of the data structures scanned and an analysis report are also produced.

UTILITY FUNCTION

The Utility function is a data administrator function. It is used to initialize the system and to perform other operations supporting the Catalogue. Through the Utility function, the data administrator can transfer entries from a working Catalogue to the permanent Catalogue, backup and restore system files, display statistics, rename entries, and renumber lines in entries.

Data Catalogue provides the user with a computerized dictionary of data processing information. The Catalogue, which contains all of this information, is created, maintained, and accessed through the use of six functions. This section describes in general the rules, requirements, and usage of Data Catalogue. Subsequent sections provide specific descriptions of entity types and individual functions.

CATALOGUE ENTRIES

Entries stored in the Catalogue are structured into categories of information. Each category contains one or more fields that can be used to define the entity type. Some fields are user defined; that is, anything meaningful can be entered in the field as long as the maximum length is not exceeded. Other fields require a one-character code that must be one of the defined codes. In most cases, a number of one-digit codes are reserved for user definition.

The categories and fields applicable to each entity type are described in section 3. The following paragraphs describe the conventions used in defining entities.

CATALOGUE NAMES

Each entry in the Catalogue is assigned a Catalogue name. This is the unique key that identifies and relates all of the data in an entry. The Catalogue name is the only field required to establish an entry in the Catalogue; however, it is recommended that at least a description be entered so that no confusion exists as to what is actually being described. The Catalogue name has a maximum length of 32 characters; spaces cannot be embedded in the name. It can consist of alphabetic characters, the digits 0 through 9, and the special characters dash and underscore.

The selection of a significant, unambiguous, and widely intelligible name is the best criteria for selecting a Catalogue name. In many cases, the COBOL preferred data name is used as the Catalogue name. This frequently encourages programmers and analysts using the Catalogue to propagate the Catalogue name more freely.

Catalogue names developed from a series of prime names, qualifiers, suffixes, and prefixes are sometimes used. This might be effective for some installations; however, carefully expressed English words are usually more readily remembered, accessed, and worked with than long and complex names. This is particularly true when creating input for Data Catalogue. The more compact the name the less work at the terminal or keypunch.

SYSTEM NAMES

Category fields are identified by system names. These names are used in entering and accessing the fields. System names must be entered exactly as shown in the individual field descriptions in section 3 and the summary of entity types in appendix D.

REPORT NAMES

The three documentation reports produced by the Report function list values stored in category fields. The names by which fields are identified on the reports are called report names. In general, the report name describes the field more clearly than the system name, which is limited to eight characters. Report names for the fields are shown in section 3 and appendix D.

CATEGORY HEADERS

The category header identifies a category of information. When an entry is being added or changed, a Category Header statement must be entered immediately preceding the field definitions for the category. In reports and listings that request detailed information, the category header is printed before the fields in that category are printed. Each category is identified by a key word such as DESCRIPTION, NAMES, or STRUCTURE.

CATEGORY LINE NUMBERS

Information in a category is stored in lines. Each line in a category is assigned a category line number by the user. Numbers in a category must be assigned in ascending sequence and must not exceed four digits. Line numbers should be assigned in increments greater than one to allow for subsequent insertions.

A category line number is the lowest accessible item within a category defined for an entity type. Some categories have only one line; the line is usually entered as line number 1. Other categories can have multiple lines; for example, the ALIAS category in an element entity definition could have one line for each alias of the element.

A category line consists of all the fields in that category. Null values are stored for fields that are not defined. When all fields being defined cannot be contained on one terminal line or one punched card, a continuation line should be used for the additional fields. If a line number is assigned to the second line, a full category line is stored for each line number with null values for fields not defined for the line number. Multiple line numbers for a single category line should be avoided because this wastes disk storage space.

CONTINUATION LINES

A continuation line is used when a category line exceeds the capacity of one terminal line or one punched card. The coding of continuation lines is as follows:

The line to be continued ends with a comma entered as the last character in the line.

The continuation line begins with a space entered as the first character in the line.

The comma in the line to be continued must immediately follow the last character of a field definition. A field system name and its definition must be on the same line. In the continuation line, the system name for a field immediately follows the space.

COMMENT LINES

Comments can be added to entries stored in the Catalogue. Inserting comments in entries might be useful in categories that contain multiple category lines. If, for example, the STRUCTURE category in a file entry specifies more than one record, comments can be included to explain the use of each record.

A comment line is indicated by an asterisk in column 1. The text of the comment follows the asterisk in free-form format. Comments are not continued from one line to the next; each comment line begins with an asterisk. The maximum length of a comment line is 67 characters including the asterisk.

Comment lines entered for an entry are stored in the Catalogue. Reports that print the entry include the stored comment lines.

PARTIAL USAGE

Procedure and user entity types can define relationships to other entries in the Catalogue; this sets up usage links. If a program, for example, is related to a file, this means that the program uses the file. In cases where the entry specifying a relationship does not use all of the related entry, partial usage can be specified for the relationship.

When an entry is related to another entry, cross-references are maintained between the entity type being defined and all the entries that comprise the related entry. For example, a program related to a record establishes cross-references to all the groups and elements defined for the record. If this relationship specifies partial usage, cross-references are maintained to only those group and element entries used by the program.

Partial usage is documented by a relational line indicating the related entry and partial usage. This is followed by a relational line for each of the subentries that is actually used. In the example of a program related to a record, the first relational line specifies the record and partial usage. Subsequent relational lines identify those groups and elements within the record that are actually used by the program.

ALIASES AND VERSIONS

An entry can be established as an alias or version of another entry in the Catalogue. An element, group, or record entity type is described as an alias; any other entity type is described as a version.

When an entry is established as an alias or version of another entry, the other entry cannot itself be an alias or version. For example, ELEMENT-A is established as an alias of ELEMENT-B; another entry can be an alias of ELEMENT-B but not ELEMENT-A.

Aliases for element entries need not be defined as separate entries. The ALIAS category in an entry can include an alias line for each alias of the element. An

alias line specifies the alias data name and its description if the description is different from the element description.

FUNCTION REQUESTS

Data Catalogue functions are invoked by requests. A function request is comprised of a series of statements that identify the function to be performed and specify the requirements for the desired operation.

The general format of a function request is described in the following paragraphs. Each function request is described in detail in the section related to the function.

REQUEST STATEMENTS

The first statement in a function request identifies the function to be performed. This is followed by one or more statements that elaborate the request. Some statements provide parameters to the requested function, some command an action, and some set options related to request processing. The function descriptions in subsequent sections identify which statements are required and optional for each function.

COMMENTS

Comments can be added to function requests input to Data Catalogue. The objective of a function request can be documented by inserting comments before a set of request statements.

A comment line is indicated by an asterisk in column 1. The text of the comment follows the asterisk in free-form format. Comments are not continued from one line to the next; each comment line begins with an asterisk. The maximum length of a comment line is 67 characters including the asterisk.

For a function request, comments should be placed before a series of statements rather than before each statement. Function request comments can be used to document the objectives of the request, to note special conditions in effect, or to explain the statements more fully.

STATEMENT CONTINUATION

Statements that exceed the capacity of one punched card or one terminal line require a continuation line. Each function request places a different limit on the number of continuation lines allowed. The coding of continuation lines is as follows:

The line to be continued ends with a comma entered as the last character in the line.

The continuation line begins with a space entered as the first character in the line.

The comma in the line to be continued must immediately follow the last character of the statement. In the continuation line, the text of the statement immediately follows the space.

An alphanumeric literal cannot be continued from one line to the next. When a large literal is part of a statement, it should be placed on a line by itself. Key words and values

also cannot be continued. A line to be continued should end at a logical division of the statement.

REQUEST FORMS

Request forms can be used to formulate function requests for either batch or on-line processing. They are useful for documenting requests in general and specifically for saving elaborate requests for re-entry at a later time.

A sample request form for each function is shown in appendix E. At the top of each form, space is provided for the name of the requester and the date. This information is for external use only. The remainder of the form is divided into sections. Each section is designed to accommodate one type of statement. A preprinted statement identifier appears at the beginning of each section; this is the first word of the statement.

The numbers 1, 72, 73, and 80 on the form represent card columns for batch input. The statement is entered in columns 1 through 72 in free-form format according to the syntax rules for the statement. Columns 73 through 80 are optional and can be used for a statement sequence number. It is recommended that sequence numbers be used for complex requests to maintain better control and to ensure that statements are submitted in the planned sequence.

ERROR MESSAGES

Function requests input to Data Catalogue are edited for syntax errors and certain data errors. Whenever Data Catalogue detects an input verification error, a diagnostic is output. A diagnostic consists of an error code and an error message. The first part of a diagnostic is an error code that contains the following information:

- A five-character code identifying the function that output the diagnostic

- An error number unique to the function

- A one-character code indicating the severity of the error

The last part of a diagnostic is an error message that explains the type of error encountered. Diagnostics are described in detail in appendix B.

When running a report or entering a command type of statement, a terse diagnostic can be output for an execution error. The function identifier, the error number, and the severity code are output. Device and input/output errors also produce highly compacted error messages.

The Catalogue is a collection of entries that describe various types of information. Each entry contains information related to one particular entity type. An entry can include descriptions, values, technical attributes, control information, usage data, structural definitions, and relationships.

Many different types of entities can be described in the Catalogue. Each entity type has a set of attributes that are appropriate to describe it. A file entity, for example, is described differently from a program entity; each has very different characteristics. Data Catalogue recognizes the following types of entities:

Element	Report
Group	External
Record	Task
File	Module
Dataset	System
Data base	User
Form	

These thirteen types of entities are described in this section. The information presented here is used for accessing and updating the Catalogue and for creating and interpreting Data Catalogue reports.

Each entity consists of various categories of information. Table 3-1 lists the available categories and indicates the entities to which the categories are applicable.

Four categories are common to all entities; these categories are described first. This is followed by a complete description of each entity. Each category applicable to the type, except the four common categories, is described separately in the entity description. The breakdown of a category is by fields. For each field, two Data Catalogue standard names are listed. System name is the name recognized by Data Catalogue during processing; that is, system name is the name used in Data Catalogue statements. Report name is the name by which a field is identified on a Data Catalogue report.

Refer to appendix D for a summary of categories and fields by entity type.

COMMON CATEGORIES

Four categories of information can be used in describing all types of entities. These categories, which are all optional, are as follows:

- CONTROL
- CLASSIFICATION
- DESCRIPTION
- OTHER

TABLE 3-1. APPLICABLE CATEGORIES FOR ENTITIES

Category	Element	Group	Record	File	Dataset	Data base	Form	Report	External	Task	Module	System	User
CONTROL	X	X	X	X	X	X	X	X	X	X	X	X	X
CLASSIFICATION	X	X	X	X	X	X	X	X	X	X	X	X	X
DESCRIPTION	X	X	X	X	X	X	X	X	X	X	X	X	X
OTHER	X	X	X	X	X	X	X	X	X	X	X	X	X
STRUCTURE		X	X	X	X	X							
RELATIONAL							X	X	X	X	X	X	X
ORIGIN	X												
VALUE	X												
ALIAS	X												
ATTRIBUTES	X	X	X	X	X		X	X			X		
RESPONSIBILITY					X	X	X	X	X	X	X	X	
NAMES	X	X	X	X	X	X	X	X	X	X	X	X	
ENVIRONMENT					X								
IOAREA						X							
UNIT				X	X								
SUPPLIERS							X						
FLOW							X						
DISTRIBUTION								X					
SCHEDULING										X	X		
LOCATION									X				
OPERATING											X		
PASSING											X		
STORAGE											X		
PERSONNEL													X

CONTROL CATEGORY

Information in the CONTROL category classifies the entry as to its status. This category is also used to document that the entry to which it applies is an alias or version of another entry that has already been defined to Data Catalogue.

Alias or Version

This field specifies that the entry is an alias or version of another entry in the Catalogue. The Catalogue name of the other entry is stored in this field. Alias is applicable to element, group, and record entities; version is applicable to all other entities. Field characteristics are as follows:

System name	EALIASOF VERSION
Report name	ALIAS OF VERSION OF
Maximum length	32 characters

Entry Status

This field contains a one-character code that indicates the present status of the entry in the Data Catalogue. Permissible codes for this field are as follows:

E	Existing entry
P	Proposed entry
O	Obsolete entry
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	ESTATUS
Report name	STATUS
Maximum length	1 character

CLASSIFICATION CATEGORY

The CLASSIFICATION category is used to store keywords for an entry. Keywords can be assigned in accordance with a structure and methodology defined by the data administrator. This allows entries to be classified according to a table of values that have meaning to a particular installation or user.

Keyword

The keywords associated with the entry are contained in this field. Query language and report selection retrieval paths can be set up for entries that share common characteristics by using keywords.

When keywords are being entered, a comma must be used to separate keywords. When entering keywords, a system name is not used. Keywords are stored in lines as text.

Each line has a maximum length of 67 characters. Multiple lines can be used without adversely impacting disk storage. The number of keywords that can be specified in an entry is determined by the field length, which is defined by the user. Field characteristics are as follows:

System name	Not applicable
Report name	Not applicable
Maximum length	User-defined value

DESCRIPTION CATEGORY

A free-form description of the entity being defined is contained in the DESCRIPTION category. Any number of lines up to 9999 can be used to describe an entity without adversely impacting disk storage.

Description

This is the only field in the DESCRIPTION category. It provides the means to define an entity in terms that are meaningful to both technical and nontechnical staff members. Each description line can contain up to 67 characters; any combination of characters can be used. Words or other values should not be continued from one line to the next line. Field characteristics are as follows:

System name	Not applicable
Report name	Not applicable
Maximum length	User-defined value

OTHER CATEGORY

Information that does not conveniently fit within the available categories can be stored in the OTHER category. Free-form comment lines can be entered to store information in this category. One specific field is provided to define COBOL level 88 entries.

COBOL Level 88

This field defines a COBOL level 88 conditional statement entry. It is a compound field and must be described syntactically correct for the COBOL compiler. Correct syntax is as follows:

condition-name = value [THRU value]

The condition name specified is a COBOL data name of up to 32 characters. The value specified is exactly what is output by the Generate function; the THRU value option is an extension of the entry. The equal sign is required. Field characteristics are as follows:

System name	88LEVEL
Report name	88 LEVEL
Maximum length	67 characters

DATA ENTITIES

Data entities are organized into hierarchies within Data Catalogue. These entity types are as follows:

- Element
- Group
- Record
- File
- Dataset
- Data base

ELEMENT ENTITY

An element is the lowest level in the data hierarchy. It is the basic building block of data structures. At the elementary level, data cannot be subdivided. Typical elements are data items such as employee number, tax code, and base pay.

In addition to the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER), the following categories can be used to describe an element:

- ORIGIN
- VALUES
- NAMES
- ATTRIBUTES
- ALIAS

ORIGIN Category

Fields in the ORIGIN category are used to describe where and when the element is originated as well as the source responsible for the element. All fields are optional. Comment lines, which begin with an asterisk, can be included in this category to provide additional information. This category provides a convenient alternate method for defining relationships between the element and users, programs, or forms.

Originating Department

This field specifies the department responsible for the element. A user unit that has not been defined to Data Catalogue can be specified. Field characteristics are as follows:

System name	DEPT
Report name	DEPARTMENT
Maximum length	32 characters

Program, Application, or System

The name of the program, application, or system originating the element is specified in this field. The program, application, or system need not be defined to Data Catalogue. Field characteristics are as follows:

System name	PROGRAM
Report name	PROGRAM
Maximum length	32 characters

Originating Form

This field documents the name or number of the form on which the element is originated. The form need not be defined to Data Catalogue. Field characteristics are as follows:

System name	FORM
Report name	FORM
Maximum length	32 characters

Generated Field Indicator

This field specifies whether the element is automatically generated by a computer program. Permissible codes for the field are as follows:

Y	Yes, the field is generated
N	No, the field is not generated
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	GENCODE
Report name	GENERATED
Maximum length	1 character

Cycle of Origin

The frequency with which the element is originated is specified by this field. Permissible codes for this field are as follows:

D	Daily
W	Weekly
M	Monthly
Q	Quarterly
S	Semiannually
A	Annually
R	On request
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	CYCLE
Report name	CYCLE
Maximum length	1 character

Estimated Life Cycle

This field specifies the estimated life cycle for the element. A numeric value is entered in this field; the unit of measure is specified by the next field described. Field characteristics are as follows:

System name	TIMELIFE
Report name	EXPECTED LIFE
Maximum length	3 digits

Estimated Life Cycle Unit

This field assigns a unit of measure code to the numeric value specified for the estimated life cycle for the element. Permissible codes for this field are as follows:

D	Days
W	Weeks
M	Months
Q	Quarters
S	Semiannual periods
A	Annual periods
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	UNITLIFE
Report name	Depends on the code selected
Maximum length	1 character

VALUES Category

The VALUES category specifies information about the content of the element being described. The values that the element can assume in its data processing life are specified. The rules by which the element is formed if it is a simple element could also be specified; for example, the element must be numeric or zero. A finite range of values can also be defined for the element.

Value

This field contains a value or editing rule applicable to the element. The value is entered in free form and is not edited. A maximum of 9999 lines of values can be entered for an element. Field characteristics are as follows:

System name	VALUE
Report name	VALUE
Maximum length	52 characters

Note to Value

A note field can be appended to each value field. It is used to footnote, amplify, or otherwise qualify the value field. Field characteristics are as follows:

System name	NOTE
Report name	NOTE
Maximum length	15 characters

NAMES Category

The element might be known by various names depending on what programming languages are used. This category is used to store the names for programs written in COBOL, PL/I, and assembler language, as well as the TOTAL data base environment.

COBOL Data Name

The name for programs written in COBOL is contained in this field. Only one COBOL data name field can be specified for the element. The data name in this field is retrieved when a COBOL Data Division entry for a record, file, or program is generated through Data Catalogue. Field characteristics are as follows:

System name	DATANAME
Report name	DATA NAME
Maximum length	32 characters

PL/I Identifier

The PL/I identifier for the element is entered in this field. The rules for forming valid PL/I identifiers must be followed when the identifier is entered. The identifier in this field is retrieved when a PL/I structure is generated through Data Catalogue. Field characteristics are as follows:

System name	IDEN
Report name	IDENTIFIER
Maximum length	31 characters

Assembler Symbol

The symbol name by which the element is known in assembler programs is entered in this field. The rules of assembly language for forming symbols must be followed when the symbol is entered. Field characteristics are as follows:

System name	SYMBOL
Report name	SYMBOL
Maximum length	8 characters

TOTAL Name

The name by which the element is known to TOTAL is entered in this field. This is the TOTAL data item name and is the name retrieved when a TOTAL data base structure is generated through Data Catalogue. Field characteristics are as follows:

System name	DBMSNAME
Report name	DBMS NAME
Maximum length	8 characters

ATTRIBUTES Category

The ATTRIBUTES category is used to define the element so that it can be described in various programming languages. This information is required if language dependent definitions are to be generated through Data Catalogue. Only one set of attribute fields can be specified unless aliases are defined for the element.

Element Length

The number of characters in the element is specified in this field. The element must be at least one character in length. Field characteristics are as follows:

System name	LENGTH
Report name	LENGTH
Maximum length	4 digits

Element Format

The format of the element value is indicated by a code entered in this field. Permissible codes for this field are as follows:

B	Binary
C	Character
F	Floating point
H	Hexadecimal
N	Numeric
O	Octal
P	Packed decimal
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	FORMAT
Report name	FORMAT
Maximum length	1 character

Justification Indicator

A code entered in this field indicates the justification of the element value. Permissible codes for this field are as follows:

L	Left-justified
R	Right-justified
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	JUST
Report name	JUSTIFIED
Maximum length	1 character

Synchronization Indicator

This field is used to indicate the synchronization of the element. Permissible codes for this field are as follows:

Y	Yes, the field is synchronized
N	No, the field is not synchronized
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	SYNC
Report name	SYNCHRONIZED
Maximum length	1 character

Volatility Indicator

The element can be classified as to its relative volatility. Elements such as employee number are relatively stable while others such as daily sales are continually changing in value. Permissible codes for this field are as follows:

D	Dynamically changing
S	Static
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	VOLATILE
Report name	VOLATILITY
Maximum length	1 character

Nature

This field is used to classify the element as to its content type; for example, dates, dollars, addresses, or quantity could be entered. It is a user-assigned unedited code; any value can be entered. Field characteristics are as follows:

System name	NATURE
Report name	NATURE
Maximum length	7 characters

Unit of Measure

This field is used to describe the unit of measure for the element; for example, gallons, pounds, or hundreds could be specified. Any value can be entered. Field characteristics are as follows:

System name	MEASURE
Report name	UNIT OF MEASURE
Maximum length	8 characters

PICTURE Clause

The COBOL language PICTURE clause for the element is specified in this field. The clause should follow the rules of COBOL, except that the word PICTURE or any abbreviation of it is not entered and the clause is not terminated by a period. Any comma in the PICTURE clause must be enclosed in quotation marks. Field characteristics are as follows:

System name	PICTURE
Report name	PICTURE
Maximum length	25 characters

Initial Value

This field is used to specify the initial value of the element when generating Data Division entries through Data Catalogue. COBOL rules for initial values must be followed except that the size of the clause is limited to 25 character positions. Data Catalogue translates this initial value to the appropriate format for generating PL/I or TOTAL entries. Field characteristics are as follows:

System name	IVALUE
Report name	INIT VALUE
Maximum length	25 characters

ALIAS Category

The ALIAS category is used to document the names and descriptions used by various programs for the same element. Elements named in this category are not described in separate entries in the Catalogue. A maximum of 9999 aliases can be entered. Each alias is entered as a separate alias line. Fields other than the alias name field are entered only when the description differs from the element entity description.

Alias Name

This field specifies the COBOL alias name for the element. COBOL rules must be followed in forming the data name. Field characteristics are as follows:

System name	ADATANAM
Report name	DATA NAME
Maximum length	32 characters

Alias Identifier, PL/I

This field specifies a PL/I alias name for the element. The rules of PL/I for valid identifiers must be followed. Field characteristics are as follows:

System name	AIDEN
Report name	IDENTIFIER
Maximum length	31 characters

Alias Symbol Name

This field specifies an assembly language alias symbol for the element. The rules of the assembly language for valid symbols must be followed. Field characteristics are as follows:

System name	ASYMBOL
Report name	SYMBOL
Maximum length	8 characters

Alias Length

If the length of the alias element differs from the length for the element entity, the length is entered in this field. The length in this field overrides the element length specified when the alias is requested for generating data definitions through Data Catalogue. Field characteristics are as follows:

System name	ALENGTH
Report name	LENGTH
Maximum length	4 digits

Alias Format

If the alias element format is different from the format for the element entity, the alias format is entered in this field. A code entered in this field overrides the element format when the alias is requested for generating data definitions through Data Catalogue. Permissible codes for this field are as follows:

B	Binary
C	Character
F	Floating point
H	Hexadecimal

- N Numeric
- O Octal
- P Packed decimal
- n User-assigned values 0 through 9

Field characteristics are as follows:

System name	AFORMAT
Report name	FORMAT
Maximum length	1 character

Alias Justification Indicator

This field specifies the justification for the alias element when it differs from the justification in the element entity definition. A code in this field overrides the element justification code when the alias is requested for generating data definitions through Data Catalogue. Permissible codes for this field are as follows:

- L Left-justified
- R Right-justified
- n User-assigned values 0 through 9

Field characteristics are as follows:

System name	AJUST
Report name	JUSTIFIED
Maximum length	1 character

Alias Synchronization Indicator

The synchronization of the alias element is entered in this field if it differs from the synchronization in the element entity definition. A code entered in this field overrides the element synchronization when the alias is requested for generating data definitions through Data Catalogue. Permissible codes for this field are as follows:

- Y Yes, the alias is synchronized
- N No, the alias is not synchronized
- n User-assigned values 0 through 9

Field characteristics are as follows:

System name	ALIASYNC
Report name	SYNCHRONIZED
Maximum length	1 character

Alias Picture

If the alias element PICTURE clause is different from the element PICTURE clause, it is entered in this field. The word PICTURE or any abbreviation is not specified; the clause is not terminated by a period. A picture entered in this field overrides the element picture when the alias is

requested for generating data definitions through Data Catalogue. Field characteristics are as follows:

System name	APICTURE
Report name	PICTURE
Maximum length	25 characters

Alias Initial Value

An initial value for the alias element is entered in this field when it differs from the element initial value. A nonnumeric value must be enclosed in quotation marks. If an initial value is specified for this field, it overrides the element initial value during Data Catalogue data definition generation requesting the alias. Field characteristics are as follows:

System name	AINIT
Report name	INIT VALUE
Maximum length	25 characters

GROUP ENTITY

A group is a logical collection of one or more elements or subgroups. Groups are used to refer to data entries collectively. The purpose in forming groups is to allow elements and subgroups to be addressed collectively as well as individually. A typical group is order date, which consists of order month, order day, and order year.

In addition to the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER), the following categories can be used to describe a group:

NAMES
ATTRIBUTES
STRUCTURE

NAMES Category

The NAMES category records the COBOL, PL/I, and assembly language group names. Only one set of names can be specified for a group. Aliases for the group must be set up using another entry.

COBOL Data Name

The COBOL data name for the group is entered in this field. The name is not edited, but it should be entered in accordance with the rules of COBOL. This name is retrieved and appended to the group when generating data definitions through Data Catalogue. Field characteristics are as follows:

System name	DATANAME
Report name	DATA NAME
Maximum length	32 characters

PL/I Identifier

The PL/I identifier for the group is entered in this field. The field is not edited, but the rules of PL/I for identifier

formation should be followed. The identifier in this field is retrieved when PL/I structures are generated through Data Catalogue. Field characteristics are as follows:

System name	IDEN
Report name	IDENTIFIER
Maximum length	31 characters

Assembler Symbol

The assembly language symbol for the group is entered in this field. The field is not edited, but the rules of assembly language for symbols should be followed. Field characteristics are as follows:

System name	SYMBOL
Report name	SYMBOL
Maximum length	8 characters

TOTAL Name

The name by which the group is known to TOTAL is entered in this field. This is the TOTAL element name. Field characteristics are as follows:

System name	DBMSNAME
Report name	DBMS NAME
Maximum length	8 characters

ATTRIBUTES Category

The group can be described in terms of its own technical aspects. Most groups are not defined at the group level because they can be defined by the elements that make up the groups.

Group Length

This field specifies the length of the group. It is the physical length of the group components. Field characteristics are as follows:

System name	LENGTH
Report name	LENGTH
Maximum length	4 digits

Group Format

The format of the group as a whole is specified in this field. Permissible codes for this field are as follows:

C	Character
B	Binary
F	Floating point
H	Hexadecimal
N	Numeric

O	Octal
P	Packed decimal
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	FORMAT
Report name	FORMAT
Maximum length	1 character

Group Picture

A PICTURE clause can be assigned to the group. The specified picture is cascaded down to each of the elements within the group. Any valid PICTURE clause can be entered. The rules of COBOL must be followed, except that the word PICTURE or any abbreviation of it is not entered and the clause is not terminated by a period. Field characteristics are as follows:

System name	PICTURE
Report name	PICTURE
Maximum length	25 characters

Group Justification

The justification of the group as a whole is specified by a code entered in this field. Permissible codes for this field are as follows:

L	Left-justified
R	Right-justified
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	JUST
Report name	JUSTIFIED
Maximum length	1 character

Group Synchronization

This field is used to indicate the synchronization of the group as a whole. Permissible codes for this field are as follows:

Y	Yes, the field is synchronized
N	No, the field is not synchronized
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	SYNC
Report name	SYNCHRONIZED
Maximum length	1 character

STRUCTURE Category

The structure of the group defines which components are included in the group and also defines the relative position of the components within the structure. Group structures can consist of a number of elements or can be multi-tiered with subordinate levels of hierarchy within the group.

The STRUCTURE category explicitly defines the group structure. Each component of the group is identified on a separate structure line by its Catalogue name. If the component is a subgroup, its structure need not be redefined.

Catalogue Name

The Catalogue name of a component in the group is entered in this field. The component must be one that already exists in the Catalogue or that has been previously created in the same update run. When this structure line is created, a cross-reference from the group being described to the lower level entry is also created. If undefined space is used in the structure, the word FILLER is entered in this field. The Catalogue name field is required when a new structure line is being entered. Field characteristics are as follows:

System name	CATNAME
Report name	CATALOGUE NAME
Maximum length	32 characters

Alias Number

This field is used when the group component named in the Catalogue name field is to be associated with the group through a specified alias of the component. If a separate entry is set up for the alias, only the Catalogue name need be entered to define the role of the component in the group structure. If the alias is stored in the ALIAS category for an element, the alias can be referenced by its alias line number. When this field is not defined, the data name, identifier, and TOTAL name are taken from the NAMES category at generation time. Field characteristics are as follows:

System name	ALIAS
Report name	ALIAS
Maximum length	4 digits

Filler Length

Undefined or unused space in the group structure is described by the filler length field. The number of character positions to be used as filler is specified in this field. A FILLER clause entry is generated for the specified number of characters when generating Data Division definitions through Data Catalogue. Field characteristics are as follows:

System name	FILLER
Report name	FILLER
Maximum length	4 digits

Redefines

The occurrence of a redefined entry within the group structure is indicated by this field. The Catalogue name of the structure line being redefined is entered in this field. The structure line must have been previously defined in the group structure. To generate correct Data Division definitions, the redefined structure should be at the same hierarchical level as the entry being redefined. Field characteristics are as follows:

System name	REDEFINE
Report name	REDEFINES
Maximum length	32 characters

Occurs From

This field is used in conjunction with the next field described to document a repeating group. If the group can occur a variable number of times, then the minimum number of times it can occur is entered in this field. If the group occurs a fixed number of times, this field is not required. Field characteristics are as follows:

System name	FROM
Report name	FROM
Maximum length	3 digits

Occurs To

This field is used to define repeating occurrences of the named component within the group. If the component occurs a fixed number of times, this field is required. If the component occurs a variable number of times, the maximum number of occurrences of the repeating component is specified in this field. Field characteristics are as follows:

System name	TO
Report name	TO
Maximum length	3 digits

Indexed By

This field is used to indicate that the group component is indexed by a data item. The data name, not the Catalogue name, of the data item is entered in this field. This data name is inserted in an INDEXED BY clause when generating Data Division entries through Data Catalogue. A standard COBOL data name should be specified for this field. Field characteristics are as follows:

System name	INDEX
Report name	INDEXED BY
Maximum length	32 characters

Depending ON

The DEPENDING ON clause for a COBOL program is documented by this field. The data name, not the Catalogue name, of the field upon which the component

depends is entered in this field. This name is moved directly into the DEPENDING ON clause when generating Data Division entries through Data Catalogue. A standard COBOL data name should be specified for this field. Field characteristics are as follows:

System name	DEPEND
Report name	DEPENDING ON
Maximum length	32 characters

Key Field

This field is used to indicate that the component listed in this structure line is a sequence key field in the group. The code in this field also indicates the type of sequence keys used. Permissible codes for this field are as follows:

D	Duplicate keys permitted
U	Unique sequential keys
M	Multiple sequential keys
N	Not a key field
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	KEY
Report name	KEY
Maximum length	1 character

Inclusion Control

The inclusion control field specifies whether the component is to be excluded when generating Data Division entries or data base definitions through Data Catalogue. Permissible codes for this field are as follows:

Y	Yes, the component is included
N	No, the component is excluded
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	INC
Report name	INCLUSION
Maximum length	1 character

RECORD ENTITY

A record is a logically related series of elements and/or groups. A record can be read from or written to a physical storage device. Groups only have existence within records. Records can be written on tape, disk, cards, or other media.

In addition to the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER), the following categories can be used to describe a record:

NAMES
ATTRIBUTES
STRUCTURE

NAMES Category

The COBOL, PL/I, and assembly language record names are stored in this category. Only one set of names can be specified for a record. Each name is used when generating language dependent data definitions through Data Catalogue.

COBOL Data Name

The COBOL data name for the record is entered in this field. The name is not edited, but it should be entered in accordance with the rules of COBOL. This name is associated with the 01 level entry in a COBOL program when generating the record through Data Catalogue. Field characteristics are as follows:

System name	DATANAME
Report name	DATA NAME
Maximum length	32 characters

PL/I Identifier

The PL/I identifier for the record is entered in this field. The name should conform to the PL/I rules for forming valid identifiers. When generating a PL/I structure through Data Catalogue, this name is appended to the structure. Field characteristics are as follows:

System name	IDEN
Report name	IDENTIFIER
Maximum length	31 characters

Assembler Symbol

The assembly language symbol for the record is stored in this field. The field is not edited, but the rules of assembly language for symbols should be followed. Field characteristics are as follows:

System name	SYMBOL
Report name	SYMBOL
Maximum length	8 characters

ATTRIBUTES Category

Only a few attributes are stored at the record level. Most of the data entries are already described as to their attributes. The attributes stored in this category relate to the record as a record.

Maximum Length

This field specifies the maximum number of characters in a record. It can be entered for records that are fixed or variable in length. Field characteristics are as follows:

System name	MAXLEN	Maximum length	32 characters
Report name	MAXIMUM LENGTH		
Maximum length	4 digits		

Minimum Length

The minimum length for variable-length records is stored in this field. Field characteristics are as follows:

System name	MINLEN
Report name	MINIMUM LENGTH
Maximum length	4 digits

PL/I Storage Attribute

The PL/I storage attribute is indicated in this field. Permissible codes for this field are as follows:

- B Based
- C Controlled
- S Static
- A Automatic
- n User-assigned values 0 through 9

Field characteristics are as follows:

System name	STORAGE
Report name	STORAGE
Maximum length	1 character

STRUCTURE Category

The structure lines in a record entity specify the components by Catalogue name and also identify their relative position within the record. The first named component is the first item in the record. A record structure can be a simple list of names of elements and/or groups that make up the record, or it can be a complex description of groups, subgroups, elements, and many variations such as repeating groups, indexed by relationships, and use of aliases. A record structure contains a separate structure line for each component in the record.

Catalogue Name

The Catalogue name of the record component is entered in this field. At least one structure line with a Catalogue name must be specified. The Catalogue name must identify an existing entry or one that has already been created in the same update run. If undefined space is used in the structure, the word FILLER is entered in this field. If the named component is made up of many subgroups and elements, only the Catalogue name of the complex group is required; the subcomponents are automatically retrieved by Data Catalogue when needed. Field characteristics are as follows:

System name	CATNAME
Report name	CATALOGUE NAME

Alias Number

This field is used when the element named in the Catalogue name field is to be associated with the record through an alias specified for the element. The alias stored in the ALIAS category for an element can be referenced by its alias line number. If a separate entry is set up for the alias, only the Catalogue name need be entered to define the role of the component in the record structure. Aliases for groups are set up as separate entry definitions with their own Catalogue names.

When this field is not defined, the data name, identifier, and TOTAL name are taken from the NAMES category at generation time. Field characteristics are as follows:

System name	ALIAS
Report name	ALIAS
Maximum length	4 digits

Filler Length

Undefined or unused space within the record structure is indicated by entering the word FILLER in the Catalogue name field and the length of the filler area in this field. A FILLER entry is generated for the specified number of characters when generating Data Division definitions through Data Catalogue. Field characteristics are as follows:

System name	FILLER
Report name	FILLER
Maximum length	4 digits

Redefines

The occurrence of a redefined entry within a record structure is indicated by this field. The Catalogue name of the structure line being redefined is entered in this field. The structure line must have been previously defined in the record structure. To generate correct Data Division definitions, the redefined structure should be at the same hierarchical level as the entry being redefined. Field characteristics are as follows:

System name	REDEFINE
Report name	REDEFINES
Maximum length	32 characters

Occurs From

This field is used in conjunction with the next field described to document a repeating group. If the group can occur a variable number of times, then the minimum number of times it can occur is entered in this field. If the group occurs a fixed number of times, this field is not required. Field characteristics are as follows:

System name	FROM
Report name	FROM
Maximum length	3 digits

Occurs To

This field is used to define repeating occurrences of the named component within the group. If the component occurs a fixed number of times, this field specifies the number of occurrences. If the component occurs a variable number of times, the maximum number of occurrences of the repeating component is specified in this field. Field characteristics are as follows:

System name	TO
Report name	TO
Maximum length	3 digits

Indexed By

This field is used to indicate that the structure line item is indexed by another data item. The data item need not be defined to Data Catalogue. The data name, not the Catalogue name, of the data item is entered in this field. This data name is inserted in an INDEXED BY clause when generating Data Division entries through Data Catalogue. A standard COBOL data name should be specified for this field. Field characteristics are as follows:

System name	INDEX
Report name	INDEXED BY
Maximum length	32 characters

Depending ON

The DEPENDING ON clause for a COBOL program is documented by this field. The data name, not the Catalogue name, of the field upon which the structure line item depends is entered in this field. This name is moved directly into the DEPENDING ON clause when generating Data Division entries through Data Catalogue. A standard COBOL data name should be specified for this field. Field characteristics are as follows:

System name	DEPEND
Report name	DEPENDING ON
Maximum length	32 characters

Key Field

This field is used to indicate that the structure line item is a sequence key field in the record. The code entered in this field also indicates the type of sequence keys used. Permissible codes for this field are as follows:

D	Duplicate keys permitted
U	Unique sequential keys
M	Multiple sequential keys
N	Not a key field
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	KEY
Report name	KEY

Maximum length	1 character
----------------	-------------

Inclusion Control

The inclusion control field specifies whether the structure line item is to be included when generating Data Division entries or data base definitions through Data Catalogue. Permissible codes for this field are as follows:

Y	Yes, the component is included
N	No, the component is excluded
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	INC
Report name	INCLUSION
Maximum length	1 character

TOTAL STRUCTURE Category

Three types of structure lines are used to describe records that occur in a TOTAL data base. The first line describes a structure item exactly as for a conventional file record using the fields just described. If the data element or item has certain characteristics, a second structure line providing three additional fields is required. If the third field in the second line is specified, a third structure line is required.

Key Field

This field indicates whether the item is a key item. Permissible codes for this field are as follows:

Y	Yes, it is a key item
N	No, it is not a key item
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	TKEY
Report name	KEY
Maximum length	1 character

Linkage Path

The master dataset name and the two characters associated with this linkage path item are entered in this field. The first four characters are the dataset name, which are followed by two characters for the identifying characters. Field characteristics are as follows:

System name	LINKPATH
Report name	LINKAGE PATH
Maximum length	6 characters

Length of Variable Portion

When the record has a variable portion and its length is

different from the length of the last named element in the structure, the total length of the variable portion is entered in this field. Field characteristics are as follows:

System name	VARLEN
Report name	LENGTH/VARIABLE
Maximum length	5 digits

When the length of variable portion field is entered, a third type of structure line is required. This line type identifies the record code associated with a particular variable portion and then specifies each of the components of the variable portion.

Record Code

The two-character user-defined code associated with the variable portion is entered in this field. Field characteristics are as follows:

System name	RCCODE
Report name	REC CODE
Maximum length	2 characters

Component Name

The Catalogue name of the component entry for the named record code is entered in this field. Each of the components should be entered on a new line for clarity. A record code field and its component name fields are followed by the next record code field and its component name fields. Field characteristics are as follows:

System name	COMPNAME
Report name	COMPONENT
Maximum length	32 characters

FILE ENTITY

A file is a collection of records. It can be described in terms of its attributes, description, structure, and relationships. Files can be stored in many different media and can be accessed in different ways.

In addition to the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER), the following categories can be used to describe a file:

RESPONSIBILITY
NAMES
ATTRIBUTES
UNIT
STRUCTURE

RESPONSIBILITY Category

The responsibility for a file is documented in this category. The history of responsibility can be traced by adding a new responsibility line each time the responsibility is reassigned; the status of the previous responsibility line can then be changed from current to past.

Status

This field specifies whether the responsibility line is for a current or past responsibility. Permissible codes for this field are as follows:

C	Current responsibility
P	Past responsibility
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	STATUS
Report name	STATUS
Maximum length	1 character

Function Code

The role of the responsible individual or unit is entered in this field. Permissible codes for this field are as follows:

D	Developed the file
C	Changed the file
T	Tested the file
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	FUNCTION
Report name	FUNCTION
Maximum length	1 character

Department

The name of the department assigned the described responsibility is entered in this field. The name entered need not be defined to Data Catalogue. Field characteristics are as follows:

System name	DEPT
Report name	DEPARTMENT
Maximum length	32 characters

Person

The name of the individual responsible for the file is entered in this field. Field characteristics are as follows:

System name	PERSON
Report name	PERSON
Maximum length	32 characters

Phone

This field contains the telephone number of the department or person responsible for the file. Either the

number and extension or the area code and number can be entered. Field characteristics are as follows:

System name	PHONE
Report name	PHONE
Maximum length	10 characters

Title

The title of the responsible individual is entered in this field. Field characteristics are as follows:

System name	TITLE
Report name	TITLE
Maximum length	15 characters

Mail Station

This field is used to document the mail station code of the responsible individual or unit. Field characteristics are as follows:

System name	MAIL
Report name	MAIL
Maximum length	5 characters

Date

The date at which the described responsibility took effect is entered in this field. Field characteristics are as follows:

System name	DATE
Report name	DATE
Maximum length	6 characters

NAMES Category

Four names for conventional files can be documented in this category.

File FD Name

The COBOL FD name for the file is entered in this field. The name must conform to COBOL conventions for FD names. Field characteristics are as follows:

System name	FDNAME
Report name	FDNAME
Maximum length	32 characters

Permanent File Name

The permanent file (or dataset) name is entered in this field. Field characteristics are as follows:

System name	DSNAME
Report name	DATA SET NAME

Maximum length	44 characters
----------------	---------------

Member Name

This field is used to specify the name of the file when it is a member of a partitioned dataset. Field characteristics are as follows:

System name	MEMBER
Report name	MEMBER NAME
Maximum length	8 characters

Logical File Name

The logical file name (or DDNAME) is entered in this field. Because there can be many different logical file names, this field is not usually practical for documenting the logical file name. Field characteristics are as follows:

System name	DDNAME
Report name	DDNAME
Maximum length	8 characters

ATTRIBUTES Category

The file can have many technical attributes. These can be defined and described by the fields in this category.

Logical Record Length

The size of one logical record is entered in this field. Field characteristics are as follows:

System name	RECSIZE
Report name	RECSIZE
Maximum length	5 digits

Block Size

The size of a block of records stored in the file is entered in this field. Field characteristics are as follows:

System name	BLKSIZE
Report name	BLKSIZE
Maximum length	5 digits

Record Format

The record format associated with the file is entered in this field. CYBER Record Manager record types can be specified. Field characteristics are as follows:

System name	RFM
Report name	RECFM
Maximum length	6 characters

File Organization

The file (or dataset) organization is specified in this field. Permissible codes for this field are as follows:

DA	Direct access
DAU	Direct access unmovable
IS	Indexed sequential
ISU	Indexed sequential unmovable
PO	Partitioned
POU	Partitioned unmovable
PS	Physical sequential
PSU	Physical sequential unmovable

Field characteristics are as follows:

System name	DSORG
Report name	DSORG
Maximum length	3 characters

File Labels

If the file is a labeled file, this field specifies the type of labels. If specified, the field is a left-justified coded field. Permissible codes for this field are as follows:

SL	Standard labels
AL	ANS labels
AUL	ANS and ANS user labels
NL	No labels
SUL	Standard and user labels
NSL	Nonstandard labels
BLP	Bypass label processing

Field characteristics are as follows:

System name	LABEL
Report name	LABEL
Maximum length	3 characters

Density

The recording density for files stored on magnetic tape is entered in this field. Permissible codes for this field are as follows:

0	200 bpi, 7-track tape
1	556 bpi, 7-track tape
2	800 bpi, 7- and 9-track tape
3	1600 bpi, 9-track tape
4	6250 bpi, 9-track tape
n	User-assigned values 5 through 9

Field characteristics are as follows:

System name	DENSITY
-------------	---------

Report name	DENSITY
-------------	---------

Maximum length	1 digit
----------------	---------

Address

The address technique for direct files is entered in this field. Permissible codes for this field are as follows:

D	Direct
I	Indirect
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	ADDRESS
Report name	ADDRESS MODE
Maximum length	1 character

Randomizing Module Name

This field is used to enter the name of the routine for accessing the file by randomizing. Field characteristics are as follows:

System name	RMODNAME
Report name	RMOD NAME
Maximum length	32 characters

Key Length

If the file has primary keys, the length of the key is entered in this field. Field characteristics are as follows:

System name	KEYLEN
Report name	KEYLEN
Maximum length	3 digits

Record Key Position

This field is used to specify the position of the primary key field. The position is relative to character position zero of the record. Field characteristics are as follows:

System name	RKP
Report name	RKP
Maximum length	3 digits

Cylinder Overflow Tracks

The number of tracks per cylinder that are to be used for cylinder overflow in an ISAM file is entered in this field. Field characteristics are as follows:

System name	CYLOFL
Report name	CYLOFL
Maximum length	2 digits

OPTCD

An option code specifying services that can be used with the access method is specified in this field. Refer to the appropriate reference manual. Field characteristics are as follows:

System name	OPTCD
Report name	OPTCD
Maximum length	3 characters

LIMCT

This field specifies a limit count. It indicates the number of blocks or tracks that are searched in an extended search operation for a direct access file. Field characteristics are as follows:

System name	LIMCT
Report name	LIMCT
Maximum length	2 digits

UNIT Category

The UNIT category is used to describe the types of devices on which the file can reside.

Unit type

A code or the model number of the device on which the file normally resides is entered in this field. A code such as DISK or a model number such as 844 could be specified. Field characteristics are as follows:

System name	PUNIT
Report name	UNIT
Maximum length	7 characters

Volume Serial Number

The volume serial number of the device on which the file resides is entered in this field. Field characteristics are as follows:

System name	PVOLSER
Report name	VOLUME SERIAL
Maximum length	6 characters

Unit Space

The amount of space that the file occupies on the specified device is entered in this field. Space requirements are expressed in various units of measure; the unit of measure for the amount in this field is specified in the next field described. Field characteristics are as follows:

System name	PSPACE
Report name	SPACE
Maximum length	3 digits

Unit Space Unit of Measure

The unit of measure for the unit space field is entered in this field. Permissible codes for this field are as follows:

B	Blocks
R	Records
C	Cylinders
T	Tracks
D	Directory blocks
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	PTYPE
Report name	Depends on the code selected
Maximum length	1 character

Index Unit

If the index for an indexed sequential file is stored on a separate unit from the file, the unit model number or generic type is entered in this field. Field characteristics are as follows:

System name	IUNIT
Report name	IDX UNIT
Maximum length	7 characters

Index Volume Serial Number

When the index for an indexed sequential file resides on a separate device, the volume serial number for that device is entered in this field. This field applies only to ISAM files. Field characteristics are as follows:

System name	IVOLSER
Report name	IDX VOLUME SERIAL
Maximum length	6 characters

Index Space Required

The amount of space required for the separate index file is specified in this field. The unit of measure for the space amount is specified by the next field described. Field characteristics are as follows:

System name	ISPACE
Report name	IDX SPACE
Maximum length	3 digits

Index Space Unit of Measure

This field specifies the unit of measure for the index space requirement. Permissible codes for this field are as follows:

B	Blocks
---	--------

R Records
 C Cylinders
 T Tracks
 D Directory Blocks
 n User-assigned values 0 through 9

Field characteristics are as follows:

System name ITYPE
 Report name Depends on the code selected
 Maximum length 1 character

Overflow Unit

When an indexed sequential file requires an overflow area that resides on a different device, the generic type or model number of the device for the overflow is entered in this field. Field characteristics are as follows:

System name OUNIT
 Report name OVFL UNIT
 Maximum length 7 characters

Overflow Volume Serial Number

The volume serial number of the overflow device is entered in this field. Field characteristics are as follows:

System name OVOLSER
 Report name OVFL VOLUME SERIAL
 Maximum length 6 characters

Overflow Space Requirements

The amount of space required to store the overflow file is entered in this field. The unit of measure for the space requirement is specified by the next field described. Field characteristics are as follows:

System name OSPACE
 Report name OVFL SPACE
 Maximum length 3 digits

Overflow Space Unit of Measure

The unit of measure for the overflow space requirements field is entered in this field. Permissible codes for this field are as follows:

B Blocks
 R Records
 C Cylinders
 T Tracks
 D Directory Blocks

n User-assigned values 0 through 9

Field characteristics are as follows:

System name OTYPE
 Report name Depends on the code selected
 Maximum length 1 character

STRUCTURE Category

The STRUCTURE category identifies the records contained in the file. Records are also classified as to header, detail, and summary records.

Catalogue Name

The Catalogue name of a record contained in the file is entered in this field. The record entry must already exist in the Catalogue or must have been previously defined in the same update run. Cross-reference entries between the file and the record are automatically established when this field is specified. Field characteristics are as follows:

System name CATNAME
 Report name CATALOGUE NAME
 Maximum length 32 characters

Record Type

The record is identified as to the type of data in the record by this field. Types such as header, detail, summary, and trailer could be specified. Field characteristics are as follows:

System name RECTYPE
 Report name TYPE
 Maximum length 8 characters

Frequency

The estimated frequency of occurrence of the record is entered in this field. Field characteristics are as follows:

System name FREQ
 Report name FREQUENCY
 Maximum length 7 digits

TOTAL DATASET ENTITY

A TOTAL dataset can be described to Data Catalogue. In addition to the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER), the following categories can be used:

RESPONSIBILITY
 NAMES
 ATTRIBUTES
 UNIT
 ENVIRONMENT
 STRUCTURE

RESPONSIBILITY Category

The responsibility for a dataset can be documented in this category. The history of responsibility can be traced by adding a new responsibility line in this category each time the responsibility is reassigned; the status of the previous responsibility line can then be changed from current to past.

Status

This field specifies whether the responsibility line is for a current or past responsibility. Permissible codes for this field are as follows:

- C Current responsibility
- P Past responsibility
- n User-assigned values 0 through 9

Field characteristics are as follows:

System name	STATUS
Report name	STATUS
Maximum length	1 character

Function Code

The role of the responsible individual or unit is entered in this field. Permissible codes for this field are as follows:

- D Developed the dataset
- C Changed the dataset
- T Tested the dataset
- n User-assigned values 0 through 9

Field characteristics are as follows:

System name	FUNCTION
Report name	FUNCTION
Maximum length	1 character

Department

The name of the department assigned the described responsibility for the dataset is entered in this field. The name entered need not be defined to Data Catalogue. Field characteristics are as follows:

System name	DEPT
Report name	DEPARTMENT
Maximum length	32 characters

Person

The name of the individual responsible for the dataset is entered in this field. Field characteristics are as follows:

System name	PERSON
Report name	PERSON

Maximum length	32 characters
----------------	---------------

Phone

This field contains the telephone number of the department or person responsible for the dataset. Either the number and extension or the area code and number can be entered. Field characteristics are as follows:

System name	PHONE
Report name	PHONE
Maximum length	10 characters

Title

The title of the responsible individual is entered in this field. Field characteristics are as follows:

System name	TITLE
Report name	TITLE
Maximum length	15 characters

Mail Station

This field is used to document the mail station code of the responsible individual or unit. Field characteristics are as follows:

System name	MAIL
Report name	MAIL
Maximum length	5 characters

Date

The date at which the described responsibility took effect is entered in this field. Field characteristics are as follows:

System name	DATE
Report name	DATE
Maximum length	6 characters

NAMES Category

Names and identifying codes associated with the dataset can be specified in this category.

TOTAL Dataset Name

The four-character name assigned to the collection of records is entered in this field. Field characteristics are as follows:

System name	TDSNAME
Report name	TOTAL DSNAME
Maximum length	4 characters

Permanent File Name

The permanent file name (or standard dataset name) is entered in this field. This is the conventional file name. Field characteristics are as follows:

System name	DSNAME
Report name	DS NAME
Maximum length	44 characters

Logical File Name

The logical file name (or DD name) is entered in this field. Field characteristics are as follows:

System name	DDNAME
Report name	DD NAME
Maximum length	8 characters

Prime Dataset Name

Where applicable, the name of the prime dataset is entered in this field. Field characteristics are as follows:

System name	PRIME
Report name	PRIME
Maximum length	8 characters

Member Name

This field is used to specify the member name of the dataset in a partitioned dataset. Field characteristics are as follows:

System name	MEMBER
Report name	MEMBER
Maximum length	8 characters

ATTRIBUTES Category

The various attributes of the TOTAL dataset can be defined as conventional file attributes by fields in this category.

Logical Record Length

The size of one logical record is entered in this field. Field characteristics are as follows:

System name	RECSIZE
Report name	RECSIZE
Maximum length	5 digits

Block Size

The size of a block of records is entered in this field. Field characteristics are as follows:

System name	BLKSIZE
-------------	---------

Report name	BLKSIZE
-------------	---------

Maximum length	5 digits
----------------	----------

Record Format

The format of the records associated with the dataset is entered in this field. Refer to the appropriate reference manuals for applicable record formats. Field characteristics are as follows:

System name	RFM
Report name	RECFM
Maximum length	6 characters

Dataset Organization

The organization of the dataset is entered in this field. Permissible codes for this field are as follows:

DA	Direct access
DAU	Direct access unmovable
IS	Indexed sequential
ISU	Indexed sequential unmovable
PO	Partitioned
POU	Partitioned unmovable
PS	Physical sequential
PSU	Physical sequential unmovable

Field characteristics are as follows:

System name	DSORG
Report name	DSORG
Maximum length	3 characters

Dataset Labels

This field specifies the type of labels for the dataset. The code must be left-justified. Permissible codes for this field are as follows:

SL	Standard labels
AL	ANS labels
AUL	ANS and ANS user labels
NL	No labels
SUL	Standard and user labels
NSL	Nonstandard labels
BLP	Bypass label processing

Field characteristics are as follows:

System name	LABEL
Report name	LABEL

Maximum length 3 characters

System name LIMCT

Report name LIMCT

Maximum length 2 digits

Randomizing Module Name

The name of the randomizing module for the dataset is entered in this field. Field characteristics are as follows:

System name RMODNAME

Report name RMOD NAME

Maximum length 32 characters

UNIT Category

This category is used to describe a dataset in terms of the kinds of device on which it can reside.

Unit Type

The generic code or model number of the device on which the dataset normally resides is entered in this field. A code such as DISK or a model number such as 844 could be specified. Field characteristics are as follows:

System name PUNIT

Report name UNIT

Maximum length 7 characters

Key Length

For a dataset containing keys, this field specifies the length of the key. Field characteristics are as follows:

System name KEYLEN

Report name KEYLEN

Maximum length 3 digits

Unit Volume Serial Number

The volume serial number of the device on which the dataset is resident is entered in this field. Field characteristics are as follows:

System name PVOLSER

Report name VOLUME SERIAL

Maximum length 6 characters

Record Key Position

The relative displacement of the key from the first record position is specified in this field. The displacement is relative to the first position, position 0. Field characteristics are as follows:

System name RKP

Report name RKP

Maximum length 3 digits

Unit Space

The amount of space that the dataset occupies on the named device is entered in this field. The unit of measure applicable to this amount is specified in the next field described. Field characteristics are as follows:

System name PSPACE

Report name SPACE

Maximum length 3 digits

Cylinder Overflow Tracks

The number of tracks per cylinder for cylinder overflow in an ISAM file is entered in this field. Field characteristics are as follows:

System name CYLOFL

Report name CYLOFL

Maximum length 2 digits

Unit Space Unit of Measure

The code entered in this field indicates the unit of measure for the unit space field. Permissible codes for this field are as follows:

- B Blocks
- R Records
- C Cylinders
- T Tracks
- D Directory blocks
- n User-assigned values 0 through 9

OPTCD

Option code services associated with the access method used by the dataset are specified in this field. Refer to the appropriate reference manual. Field characteristics are as follows:

System name OPTCD

Report name OPTCD

Maximum length 3 characters

LIMCT

A limit count for the number of blocks or tracks that are searched in an extended search operation is entered in this field. This is applicable only to direct accessing datasets. Field characteristics are as follows:

Field characteristics are as follows:

System name	PTYPE
Report name	Depends on the code selected
Maximum length	1 character

Index Unit

This field is used when the index for an indexed sequential file is stored on a separate unit. The unit model number or generic type is entered in this field. Field characteristics are as follows:

System name	IUNIT
Report name	IDX UNIT
Maximum length	7 characters

Index Volume Serial Number

The volume serial number of an index that resides on a separate unit is entered in this field. This field applies only to ISAM datasets. Field characteristics are as follows:

System name	IVOLSER
Report name	IDX VOLUME SERIAL
Maximum length	6 characters

Index Space Required

The amount of space required to store the separate index dataset is entered in this field. The unit of measure for the amount specified in this field is indicated by the next field described. Field characteristics are as follows:

System name	ISPACE
Report name	IDX SPACE
Maximum length	3 digits

Index Space Unit of Measure

The unit of measure for the amount of space for the separate index dataset is entered in this field. Permissible codes for this field are as follows:

- B Blocks
- R Records
- C Cylinders
- T Tracks
- D Directory Blocks
- n User-assigned values 0 through 9

Field characteristics are as follows:

System name	ITYPE
Report name	Depends on the code selected
Maximum length	1 character

Overflow Unit

If the indexed sequential dataset requires an overflow area that resides on a different device, the generic code or model number for the overflow device is entered in this field. Field characteristics are as follows:

System name	OUNIT
Report name	OVFL UNIT
Maximum length	7 characters

Overflow Volume Serial Number

The volume serial number of the overflow device is entered in this field. Field characteristics are as follows:

System name	OVOLSER
Report name	OVFL VOLUME SERIAL
Maximum length	6 characters

Overflow Space Requirements

The amount of space required to store the overflow dataset is entered in this field. The next field described is used to indicate the unit of measure applicable to the amount specified. Field characteristics are as follows:

System name	OSPACE
Report name	OVFL SPACE
Maximum length	3 digits

Overflow Space Unit of Measure

The unit of measure for the overflow space required is specified in this field. Permissible codes for this field are as follows:

- B Blocks
- R Records
- C Cylinders
- T Tracks
- D Directory Blocks
- n User-assigned values 0 through 9

Field characteristics are as follows:

System name	OTYPE
Report name	Depends on the code selected
Maximum length	1 character

ENVIRONMENT Category

The physical environment of a TOTAL dataset is described by this category. The amount of records that can be loaded is also indicated.

Device

The device used to support the dataset is entered in this field. A device type (such as 844 or 841) or a DASD (such as 3330 or TELEXDD) is specified. Field characteristics are as follows:

System name	DEVICE
Report name	DEVICE
Maximum length	7 characters

Total Records

The total number of records to be stored in the dataset is entered in this field. Field characteristics are as follows:

System name	TRECORDS
Report name	TOTAL RECORDS
Maximum length	7 digits

Total Tracks

This field specifies the total number of tracks that can be used in this dataset. Field characteristics are as follows:

System name	TTRACKS
Report name	TOTAL TRACKS
Maximum length	5 digits

Logical Record Length

The length of one logical record is entered in this field. Field characteristics are as follows:

System name	TLENGTH
Report name	RECD LENGTH
Maximum length	5 digits

Block Size

The number of records per block (or blocks per track) is entered in this field. Field characteristics are as follows:

System name	BLOCKS
Report name	BLOCKS/TRACK
Maximum length	5 digits

Disk Extents

The number of disk extents available is specified in this field. Field characteristics are as follows:

System name	TEXTENTS
Report name	# OF EXTENTS
Maximum length	2 digits

Cylinder Load Limit

This field specifies the percentage limit to which cylinders can be loaded. Field characteristics are as follows:

System name	LOADPCT
Report name	LOAD % LIMIT
Maximum length	2 digits

Old File

This field is used by CONVERT 7 users to specify whether an old file exists. Permissible codes for this field are as follows:

Y	Yes, an old file exists
N	No, an old file does not exist
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	OLDFILE
Report name	OLD FILE
Maximum length	1 character

Relative Start Track

This field is used to specify the starting track number. Field characteristics are as follows:

System name	RELSTART
Report name	REL START
Maximum length	5 digits

VTOC Tracks

The number of tracks that have been allocated is specified in this field. Field characteristics are as follows:

System name	VTOCTRKS
Report name	VTOC TRACKS
Maximum length	5 digits

STRUCTURE Category

The records that can be stored in the dataset are identified in this category. A record structure line can also classify the record as to its type.

Catalogue Name

The Catalogue name of a dataset record is entered in this field. The record must be defined to Data Catalogue before it is specified or have been previously defined in the same update run. When the record name is entered in this field, cross-reference entries between the file and the record are set up. Field characteristics are as follows:

System name CATNAME
 Report name CATALOGUE NAME
 Maximum length 32 characters

Record Type

The dataset record is identified as to the type of data in the record by this field. Types such as header, detail, summary, and trailer could be specified. Field characteristics are as follows:

System name RECTYPE
 Report name TYPE
 Maximum length 8 characters

Frequency

The estimated frequency of occurrence of the record type is entered in this field. Field characteristics are as follows:

System name FREQ
 Report name FREQUENCY
 Maximum length 7 digits

TOTAL DATA BASE ENTITY

A TOTAL data base can be described to Data Catalogue. In addition to the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER), the following categories can be used:

RESPONSIBILITY
 NAMES
 IOAREA
 STRUCTURE

RESPONSIBILITY Category

The responsibility for a TOTAL data base is documented in this category. The history of responsibility can be traced by adding a new responsibility line in this category each time the responsibility is reassigned; the status of the previous responsibility line can then be changed from current to past.

Status

This field specifies whether the responsibility line is for a current or past responsibility. Permissible codes for this field are as follows:

C Current responsibility
 P Past responsibility
 n User-assigned values 0 through 9

Field characteristics are as follows:

System name STATUS
 Report name STATUS
 Maximum length 1 character

Function Code

The role of the responsible individual or unit is entered in this field. Permissible codes for this field are as follows:

D Developed the TOTAL data base
 C Changed the TOTAL data base
 T Tested the TOTAL data base
 n User-assigned values 0 through 9

Field characteristics are as follows:

System name FUNCTION
 Report name FUNCTION
 Maximum length 1 character

Department

The name of the department assigned the described responsibility for the TOTAL data base is entered in this field. The name entered need not be defined to Data Catalogue. Field characteristics are as follows:

System name DEPT
 Report name DEPARTMENT
 Maximum length 32 characters

Person

The name of the individual responsible for the TOTAL data base is entered in this field. Field characteristics are as follows:

System name PERSON
 Report name PERSON
 Maximum length 32 characters

Phone

This field contains the telephone number of the department or person responsible for the TOTAL data base. Either the number and extension or the area code and number can be entered. Field characteristics are as follows:

System name PHONE
 Report name PHONE
 Maximum length 10 characters

Title

The title of the responsible individual is entered in this field. Field characteristics are as follows:

System name TITLE
 Report name TITLE
 Maximum length 15 characters

Mail Station

This field is used to document the mail station code of the responsible individual or unit. Field characteristics are as follows:

System name	MAIL
Report name	MAIL
Maximum length	5 characters

Date

The date at which the described responsibility took effect is entered in this field. Field characteristics are as follows:

System name	DATE
Report name	DATE
Maximum length	6 characters

NAMES Category

Identifying information for the TOTAL data base can be documented in this category.

TOTAL Data Base Name

The name by which the data base is known to TOTAL is entered in this field. Field characteristics are as follows:

System name	DBNAME
Report name	DB NAME
Maximum length	8 characters

Known As

The name by which the data base is generally known is entered in this field. Field characteristics are as follows:

System name	NAME
Report name	NAME
Maximum length	32 characters

IOAREA Category

The attributes of IO areas used in the TOTAL data base are defined by fields in this category.

Track Hold Code

This field indicates whether the data base is to make use of the track hold option. Permissible codes for this field are as follows:

Y	Yes, the track hold option is used
N	No, the track hold option is not used
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	TRACK
Report name	TRACK HOLD
Maximum length	1 character

Define IOAREA and Occurrences (Prologue Section)

For each defined IOAREA, an IOAREA line is specified with two fields defined; one is for the name and one is for the number of occurrences of that IOAREA. Two fields are entered in this line.

IOAREA Internal Name

The four-byte TOTAL name of the IOAREA is entered in this field. Field characteristics are as follows:

System name	IOAREA
Report name	IOAREA NAME
Maximum length	4 characters

Number of Occurrences

The number of occurrences of the IOAREA is entered in this field. Field characteristics are as follows:

System name	OCCURS
Report name	OCCURS
Maximum length	2 digits

STRUCTURE Category

The structure of the TOTAL data base is described in this category.

Dataset Name

The Catalogue name of the dataset that is a part of the TOTAL data base being described is entered in this field. Field characteristics are as follows:

System name	DSNAME
Report name	DATASET NAME
Maximum length	32 characters

Dataset Type

The type of the dataset is specified by a code in this field. Permissible codes for this field are as follows:

M	Master dataset
V	Variable dataset
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	TYPE
Report name	TYPE
Maximum length	1 character

Exclude

This field is used to indicate whether the dataset is excluded when the Generate function is used. Permissible codes for this field are as follows:

Y	Yes, exclude the dataset
N	No, do not exclude the dataset

If this field is not specified, the dataset is generated in DBDL. Field characteristics are as follows:

System name	EXCLUDE
Report name	EXCLUDE
Maximum length	1 character

PROCEDURE ENTITIES

Procedure entities have relationships between themselves; they can also interact with data entity definitions or responsibility entries. These entities are used to describe the processes that perform a task and the resources sometimes required to complete the task. Procedure entities are as follows:

- Forms
- Reports
- External resources
- Programs and modules
- Manual tasks
- Systems

FORM ENTITY

A form is a printed or typed document that includes spaces for the insertion of data. Input data for computer programs is often prepared on forms and output data from computer programs is often printed on forms. Forms play an important information processing role in the beginning, ending, and intermediate points of the cycle.

In addition to the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER), the following categories can be used to describe a form:

- RESPONSIBILITY
- NAMES
- ATTRIBUTES
- SUPPLIERS
- FLOW
- RELATIONAL

RESPONSIBILITY Category

The responsibility for a form is documented in this category. The history of responsibility can be traced by adding a new responsibility line each time the responsibility is reassigned; the status of the previous responsibility line can then be changed from current to past.

Status

This field specifies whether the responsibility line is for a current or past responsibility. Permissible codes for this field are as follows:

C	Current responsibility
P	Past responsibility
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	STATUS
Report name	STATUS
Maximum length	1 character

Function Code

The role of the responsible individual or unit is entered in this field. Permissible codes for this field are as follows:

D	Developed the form
C	Changed the form
T	Tested the form
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	FUNCTION
Report name	FUNCTION
Maximum length	1 character

Department

The name of the department assigned the described responsibility for the form is entered in this field. The name entered need not be defined to Data Catalogue. Field characteristics are as follows:

System name	DEPT
Report name	DEPARTMENT
Maximum length	32 characters

Person

The name of the individual responsible for the form is entered in this field. Field characteristics are as follows:

System name	PERSON
Report name	PERSON
Maximum length	32 characters

Phone

This field contains the telephone number of the department or person responsible for the form. Either the number and extension or the area code and number can be entered. Field characteristics are as follows:

System name PHONE
Report name PHONE
Maximum length 10 characters

Maximum length 10 characters

ATTRIBUTES Category

The attributes of the form are described in this category in terms of the number of parts and the color of the form.

Title

The title of the responsible individual is entered in this field. Field characteristics are as follows:

System name TITLE
Report name TITLE
Maximum length 15 characters

Parts

The number of parts in the form is entered in this field. Field characteristics are as follows:

System name PARTS
Report name PARTS
Maximum length 2 digits

Mail Station

This field is used to document the mail station code of the responsible individual or unit. Field characteristics are as follows:

System name MAIL
Report name MAIL
Maximum length 5 characters

Color

The color of the form is entered in this field. Field characteristics are as follows:

System name COLOR
Report name COLOR
Maximum length 10 characters

Date

The date at which the described responsibility took effect is entered in this field. Field characteristics are as follows:

System name DATE
Report name DATE
Maximum length 6 characters

SUPPLIERS Category

All the alternative suppliers that can provide the form can be documented in this category. Each supplier line provides a source for determining a supplier if any difficulties arise.

NAMES Category

The common name and number of the form are documented in this category.

Form Name

The title name of the form is entered in this field. This should be the name by which the form is generally known; it need not be the Catalogue name. Field characteristics are as follows:

System name NAME
Report name FORM NAME
Maximum length 32 characters

Supplier Name

The name of the supplier of the form is entered in this field. Field characteristics are as follows:

System name SNAME
Report name SUPPLIER NAME
Maximum length 32 characters

Supplier Number

The vendor number of the supplier is entered in this field. Field characteristics are as follows:

System name SUPNO
Report name SUPPLIER NUMBER
Maximum length 10 characters

Form Number

The form number assigned to the form is entered in this field. Field characteristics are as follows:

System name NUMBER
Report name FORM NUMBER

Reorder Point

This field is used to record the point at which new forms must be ordered. The unit of measure for the specified quantity is entered in the next field described. Field characteristics are as follows:

System name QREORDER
Report name REORDER POINT
Maximum length 5 digits

Reorder Point Unit of Measure

The unit of measure applicable to the reorder quantity is entered in this field. Permissible codes for this field are as follows:

B Box
C Carton
H Hundred
R Ream
T Thousand
n User-assigned values 0 through 9

Field characteristics are as follows:

System name UREORDER
Report name Depends on the code selected
Maximum length 1 character

On Hand Quantity

The quantity of forms on hand is entered in this field. The next field described is used to indicate the unit of measure for the quantity on hand. Field characteristics are as follows:

System name QONHAND
Report name ONHAND QTY
Maximum length 5 digits

On Hand Quantity Unit of Measure

The unit of measure for the quantity on hand is entered in this field. Permissible codes for this field are as follows:

B Box
C Carton
H Hundred
R Ream
T Thousand
n User-assigned values 0 through 9

Field characteristics are as follows:

System name UONHAND
Report name Depends on the code selected
Maximum length 1 character

Economic Order Quantity

This field is used to enter the economic quantity of forms to be ordered. The unit of measure for this field is indicated in the next field described. Field characteristics are as follows:

System name GEOQ
Report name EOQ
Maximum length 5 digits

Economic Order Quantity Unit of Measure

The unit of measure for the economic order quantity field is entered in this field. Permissible codes for this field are as follows:

B Box
C Carton
H Hundred
R Ream
T Thousand
n User-assigned values 0 through 9

Field characteristics are as follows:

System name UEOQ
Report name Depends on the code selected
Maximum length 1 character

Minimum Order Quantity

If the minimum order quantity is different from the economic order quantity, that quantity is entered in this field. The unit of measure for the minimum order quantity is entered in the next field described. Field characteristics are as follows:

System name QMINORD
Report name MINIMUM ORDER QUANTITY
Maximum length 5 digits

Minimum Order Quantity Unit of Measure

The unit of measure for the minimum order quantity is entered in this field. Permissible codes for this field are as follows:

B Box
C Carton
H Hundred
R Ream
T Thousand
n User-assigned values 0 through 9

Field characteristics are as follows:

System name	UMINORD
Report name	Depends on the code selected
Maximum length	1 character

Supplier Lead Time

The lead time required by the supplier for the form is entered in this field. The unit of measure for the lead time is specified by the next field described. Field characteristics are as follows:

System name	STIME
Report name	LEAD TIME
Maximum length	3 digits

Supplier Lead Time Unit of Measure

The unit of measure for the supplier lead time field is entered in this field. Permissible codes for this field are as follows:

D	Days
W	Weeks
M	Months
Q	Quarters
A	Annual periods
S	Semiannual periods
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	SUNIT
Report name	Depends on the code selected
Maximum length	1 character

FLOW Category

The flow of the form from department to department can be documented in this category. This documentation is extremely useful in determining work-flow dependencies.

Flow Code

Each form has a starting point, an ending point, and intermediate processing steps. This field is used to specify the nature of the work done with the form by the specified department. Permissible codes for this field are as follows:

O	Originated the form
P	Processed the form
U	Used the form as an end use
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	FLOWCODE
Report name	FLOW
Maximum length	1 character

Department

The name of the department that is processing, using, or originating the form is entered in this field. If the department has been defined to Data Catalogue, the Catalogue name can be specified. Field characteristics are as follows:

System name	FDEPT
Report name	DEPARTMENT
Maximum length	32 characters

Person

The name of the individual working with the form is entered in this field. Field characteristics are as follows:

System name	FPERSON
Report name	PERSON
Maximum length	32 characters

Phone

The telephone number of the individual or department using the form is entered in this field. The number and extension or the area code and number can be specified. Field characteristics are as follows:

System name	FPHONE
Report name	PHONE
Maximum length	10 characters

Parts Completed

This field is used to specify the number of parts of the form that are completed by the department. Field characteristics are as follows:

System name	FPARTS
Report name	PARTS
Maximum length	2 digits

Submission Day

The day or days on which the form is submitted for processing is entered in this field. Field characteristics are as follows:

System name	FSUBDAY
Report name	DAY
Maximum length	10 characters

RELATIONAL Category

The relationship between the form and other entries in the Catalogue is documented in this category. The form can be linked to a management unit, file, program, report, display, or manual task. Establishing relationships sets up linkage paths to access the form or other entries, one from the other. The cross-referencing ability allows the initial data preparation on the form to be linked to the various information processing cycles. Elements that are used in the form can be defined.

Related Entry Name

The Catalogue name of the entry to which the form is related is entered in this field. This field is required if a relationship is to be established. Refer to table 3-2 for valid relational entity types. Field characteristics are as follows:

System name	CATNAME
Report name	CATALOGUE NAME
Maximum length	32 characters

Partial Usage Code

This field is used to indicate that the form only uses part of the related entry. If the related entry is a record, for example, the form might use only three fields from the record. By specifying partial usage in this field, links are not created to all the fields in that record. The form could be related to each of the three fields instead of to the record. Permissible codes for this field are as follows:

Y	Yes, partial use only
N	No, the complete entry is related
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	PARTIAL
Report name	PARTIAL
Maximum length	1 character

REPORT (SCREEN) ENTITY

A report (or screen) is a formatted presentation of data that is prepared either by computer or manually. A report often provides the end product of the information cycle. Full use of relational capabilities can link a report to the entire information processing cycle.

In addition to the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER), the following categories can be used to describe a report:

RESPONSIBILITY
NAMES
ATTRIBUTES
FLOW
RELATIONAL

RESPONSIBILITY Category

The responsibility for a report can be documented in this category. The history of responsibility can be traced by adding a new responsibility line each time the responsibility is reassigned; the status of the previous responsibility line can then be changed from current to past.

Status

This field specifies whether the responsibility line is for a current or past responsibility. Permissible codes for this field are as follows:

C	Current responsibility
P	Past responsibility
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	STATUS
Report name	STATUS
Maximum length	1 character

TABLE 3-2. VALID ENTITY RELATIONSHIPS

Entity Type	System	Module	Task	Report	Form	External	File	Dataset	Data Base	Record	Group	Element
USER	X	X	X	X	X	X	X	X	X	X	X	X
SYSTEM	X	X	X	X	X	X	X	X	X	X	X	X
MODULE		X		X	X	X	X	X	X	X	X	X
TASK			X	X	X	X	X	X	X	X	X	X
REPORT				X	X	X	X	X	X	X	X	X
FORM				X	X	X	X	X	X	X	X	X
EXTERNAL				X	X	X	X	X	X	X	X	X
FILE										X	X	X
DATASET										X	X	X
DATA BASE								X		X	X	X
RECORD										X	X	X
GROUP											X	X

Function Code

The role of the responsible individual or unit is entered in this field. Permissible codes for this field are as follows:

D	Developed the report
C	Changed the report
T	Tested the report
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	FUNCTION
Report name	FUNCTION
Maximum length	1 character

Department

The name of the department assigned the described responsibility for the report is entered in this field. The name entered need not be defined to Data Catalogue. Field characteristics are as follows:

System name	DEPT
Report name	DEPARTMENT
Maximum length	32 characters

Person

The name of the individual responsible for the report is entered in this field. Field characteristics are as follows:

System name	PERSON
Report name	PERSON
Maximum length	32 characters

Phone

This field contains the telephone number of the department or person responsible for the report. Either the number and extension or the area code and number can be entered. Field characteristics are as follows:

System name	PHONE
Report name	PHONE
Maximum length	10 characters

Title

The title of the responsible individual is entered in this field. Field characteristics are as follows:

System name	TITLE
Report name	TITLE
Maximum length	15 characters

Mail Station

This field is used to document the mail station code of the responsible individual or unit. Field characteristics are as follows:

System name	MAIL
Report name	MAIL
Maximum length	5 characters

Date

The date at which the described responsibility took effect is entered in this field. Field characteristics are as follows:

System name	DATE
Report name	DATE
Maximum length	6 characters

NAMES Category

The names by which the report is generally known can be entered in this field. The report number and the form used for printing the report can also be specified.

Report Name

This field is used to document the name by which the report is generally known. Field characteristics are as follows:

System name	NAME
Report name	REPORT NAME
Maximum length	32 characters

Report Number

If the report is numbered, this field is used to document the report number. Field characteristics are as follows:

System name	NUMBER
Report name	REPORT NUMBER
Maximum length	10 characters

Form Name

The name of the form on which the report is printed in entered in this field. Field characteristics are as follows:

System name	FORMNAME
Report name	FORM NAME
Maximum length	32 characters

ATTRIBUTES Category

The attributes of the report can be documented in this category. The medium on which the report is produced and the number of parts to the report can be specified.

Medium

This field is entered to indicate the medium on which the report is presented. Permissible codes for this field are as follows:

D	Display
M	Microfile
P	Printed
S	Spooled
T	Typed
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	MEDIUM
Report name	MEDIUM
Maximum length	1 character

Number of Parts

The number of parts printed for a multiple copy form is entered in this field. Field characteristics are as follows:

System name	PARTS
Report name	PARTS
Maximum length	2 digits

FLOW Category

The flow of the report through the information processing cycle can be documented in this category. Each step in the life of a report can be entered in a series of flow lines that pinpoint the routing and action taken at each step.

Department

The name of the department performing the actions described in the flow line is entered in this field. The department can, but need not, be a management unit defined to Data Catalogue. Field characteristics are as follows:

System name	FDEPT
Report name	DEPARTMENT
Maximum length	32 characters

Person

This field is used to identify the individual involved with the report. Field characteristics are as follows:

System name	FPERSON
-------------	---------

Report name	PERSON
Maximum length	32 characters

Phone

The telephone number of the individual or department working on the report is entered in this field. The number and extension or the area code and number can be specified. Field characteristics are as follows:

System name	FPHONE
Report name	PHONE
Maximum length	10 characters

Number of Copies

The number of copies of the report received by the individual or department is entered in this field. Field characteristics are as follows:

System name	FCOPIES
Report name	COPIES
Maximum length	2 digits

Report Mode

This field is used to identify the operations that are performed on a hard copy report. Permissible codes for this field are as follows:

B	Burst
D	Decollated
M	Microfiche
F	Fastened
R	Reduced
n	User-assigned values 0 through 9

Up to five operations can be specified for a flow line. Field characteristics are as follows:

System name	F1MODE F2MODE F3MODE F4MODE F5MODE
Report name	MODE: 1 MODE: 2 MODE: 3 MODE: 4 MODE: 5
Maximum length	1 character

Distribution Day

The day of the week when the individual or department receives the report is entered in this field. Field characteristics are as follows:

System name FDISTDAY
 Report name DAY
 Maximum length 10 characters

Retention Period

The retention period for a report is entered in this field. The unit of measure applicable to this field is specified by the next field described. Field characteristics are as follows:

System name FTRETAIN
 Report name RETENTION
 Maximum length 3 digits

Retention Period Unit of Measure

The unit of measure for the retention period field is entered in this field. Permissible codes for this field are as follows:

- D Days
- W Weeks
- M Months
- Q Quarters
- S Semiannual periods
- A Annual periods
- I Indefinitely
- n User-assigned values 0 through 9

Field characteristics are as follows:

System name FURETAIN
 Report name Depends on the code selected
 Maximum length 1 character

RELATIONAL Category

This category is used to relate the report or screen to other entries in the Catalogue. The act of creating relationships sets up cross-referencing abilities between the named entries. A report can be related to any other entry including other reports, records, programs, systems, manual tasks, or management units. A report could be related to all the data elements that appear on it as well as to the program that produces it.

Related Entry Name

This field is used to specify the Catalogue name of the entry to which the report is related. The Catalogue name must be one defined to Data Catalogue. This field is required to specify a relationship. Refer to table 3-2 for

valid relational entity types. Field characteristics are as follows:

System name CATNAME
 Report name CATALOGUE NAME
 Maximum length 32 characters

Partial Use Code

A code is entered in this field to indicate whether all or part of the entry is related to the report. Permissible codes for this field are as follows:

- Y Yes, partial use only
- N No, the complete entry is related
- n User-assigned values 0 through 9

Field characteristics are as follows:

System name PARTIAL
 Report name PARTIAL
 Maximum length 1 character

Line Type Code

The type of report line being described is specified by this field. A line type such as header, detail, or summary can be entered. Field characteristics are as follows:

System name LTYPE
 Report name TYPE
 Maximum length 8 characters

Edit PICTURE Clause

The PICTURE clause for describing the report line entry is entered in this field. The picture specified should conform to the COBOL rules for a PICTURE clause, except that the word PICTURE is not specified and it is not terminated by a period. Field characteristics are as follows:

System name PICTURE
 Report name PICTURE
 Maximum length 25 characters

Starting Position

This field is used when each field within a report line is being related to an entry. The high-order position of the report line, relative to print position 1, is specified for the starting position. Field characteristics are as follows:

System name START
 Report name START
 Maximum length 3 digits

Ending Position

This field is used in conjunction with the previous field described to define the positioning of the report field on the report line. The relative distance from print position 1 to the low-order report location is entered in this field. Field characteristics are as follows:

System name	END
Report name	END
Maximum length	3 digits

EXTERNAL RESOURCE ENTITY

An external resource is anything used in the information processing cycle that is not computer accessible. Typical external resources are data entry instructions, system flowcharts, and an externally maintained table of values. An external resource can be related to other entries. This provides the means to track all of the tools and resources that are used in the information processing function.

In addition to the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER), the following categories can be used to describe an external resource:

RESPONSIBILITY
NAMES
LOCATION
RELATIONAL

RESPONSIBILITY Category

The responsibility for an external resource can be documented in this category. The history of responsibility can be traced by adding a new responsibility line in this category each time the responsibility is reassigned; the status of the previous responsibility line can then be changed from current to past.

Status

This field specifies whether the responsibility line is for a current or past responsibility. Permissible codes for this field are as follows:

C	Current responsibility
P	Past responsibility
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	STATUS
Report name	STATUS
Maximum length	1 character

Function Code

The role of the responsible individual or unit is entered in this field. Permissible codes for this field are as follows:

D	Developed the resource
C	Changed the resource
T	Tested the resource
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	FUNCTION
Report name	FUNCTION
Maximum length	1 character

Department

The name of the department assigned the described responsibility for the external resource is entered in this field. The name entered need not be defined to Data Catalogue. Field characteristics are as follows:

System name	DEPT
Report name	DEPARTMENT
Maximum length	32 characters

Person

The name of the individual responsible for the external resource is entered in this field. Field characteristics are as follows:

System name	PERSON
Report name	PERSON
Maximum length	32 characters

Phone

This field contains the telephone number of the department or person responsible for the external resource. Either the number and extension or the area code and number can be entered. Field characteristics are as follows:

System name	PHONE
Report name	PHONE
Maximum length	10 characters

Title

The title of the responsible individual is entered in this field. Field characteristics are as follows:

System name	TITLE
Report name	TITLE
Maximum length	15 characters

Mail Station

This field is used to document the mail station code of the responsible individual or unit. Field characteristics are as follows:

System name	MAIL
Report name	MAIL
Maximum length	5 characters

Date

The date at which the described responsibility took effect is entered in this field. Field characteristics are as follows:

System name	DATE
Report name	DATE
Maximum length	6 characters

NAMES Category

An identifying name and number for an external resource can be specified in this category.

External Resource Name

The name by which the external resource is generally known is entered in this field. Field characteristics are as follows:

System name	NAME
Report name	RESOURCE NAME
Maximum length	32 characters

Identifying Number

This field is used to specify an identifying number associated with the external resource. Field characteristics are as follows:

System name	ID
Report name	RESOURCE ID
Maximum length	10 characters

LOCATION Category

The type and location of the external resource can be documented in this category.

Resource Type

This field is used to identify the type of external resource being described. Permissible codes for this field are as follows:

D	Documentation
F	Flowchart

O	Operating instructions
L	Listing
P	Procedural instructions
S	Status report
T	Test data
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	TYPRES
Report name	RESOURCE TYPE
Maximum length	1 character

Resource Location

The location of the external resource is entered in this field. The location can be in the form of an address, substation, file location, or any other location description. This field is a free-form field. Field characteristics are as follows:

System name	LOCRES
Report name	LOCATION
Maximum length	66 characters

RELATIONAL Category

An external resource can be related to other Data Catalogue entities. A program listing or flowchart could be related to its program; a data entry procedural instruction could be related to a transaction file.

Related Entry Name

The Catalogue name of the entry to which the external resource is related is entered in this field. The name must identify an existing Catalogue entry. This field must be specified to define a relationship. Refer to table 3-2 for valid relational entity types. Field characteristics are as follows:

System name	CATNAME
Report name	CATALOGUE NAME
Maximum length	32 characters

Partial Use Code

This field is used to specify whether all or part of the named entry is related to the external resource. Additional relational lines can be used to indicate precisely which portions of the entry are related. Permissible codes for this field are as follows:

Y	Yes, partial use only
N	No, the complete entry is related
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	PARTIAL
Report name	PARTIAL
Maximum length	1 character

PROGRAM (MODULE) ENTITY

A computer program or module can be described to Data Catalogue. In the following text the word module implies program or module. A module can also be related to other entries.

In addition to the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER), the following categories can be used to describe a module:

RESPONSIBILITY
NAMES
ATTRIBUTES
STORAGE
OPERATING
SCHEDULING
PASSING
RELATIONAL

RESPONSIBILITY Category

The responsibility for a module can be documented in this category. The history of responsibility can be traced by adding a new responsibility line each time the responsibility is reassigned; the status of the previous responsibility line can then be changed from current to past. As the module is designed, tested, and later changed, the responsibility for these activities can be documented. This facilitates the management and control of the module.

Status

This field specifies whether the responsibility line is for a current or past responsibility. Permissible codes for this field are as follows:

C	Current responsibility
P	Past responsibility
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	STATUS
Report name	STATUS
Maximum length	1 character

Function Code

The role of the responsible individual or unit is entered in this field. The code in this field indicates whether the responsibility is for development, testing, or maintenance. Permissible codes for this field are as follows:

D	Developed the module
---	----------------------

C	Changed the module
T	Tested the module
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	FUNCTION
Report name	FUNCTION
Maximum length	1 character

Department

The name of the department assigned the responsibility for the module is entered in this field. The name entered need not be defined to Data Catalogue. Field characteristics are as follows:

System name	DEPT
Report name	DEPARTMENT
Maximum length	32 characters

Person

The name of the individual responsible for the module is entered in this field. Field characteristics are as follows:

System name	PERSON
Report name	PERSON
Maximum length	32 characters

Phone

This field contains the telephone number of the department or person responsible for the module. Either the number and extension or the area code and number can be entered. Field characteristics are as follows:

System name	PHONE
Report name	PHONE
Maximum length	10 characters

Title

The title of the responsible individual is entered in this field. Field characteristics are as follows:

System name	TITLE
Report name	TITLE
Maximum length	15 characters

Mail Station

This field is used to document the mail station code of the responsible individual or unit. Field characteristics are as follows:

System name MAIL
 Report name MAIL
 Maximum length 5 characters

R RPG
 n User-assigned values 0 through 9

Date

The date at which the described responsibility took effect is entered in this field. Field characteristics are as follows:

System name DATE
 Report name DATE
 Maximum length 6 characters

Field characteristics are as follows:

System name LANGPRG
 Report name LANGUAGE
 Maximum length 1 character

NAMES Category

The name and identifying number of the module can be specified in this category.

Module Name

The name by which the module is generally known is entered in this field. Field characteristics are as follows:

System name NAME
 Report name MODULE NAME
 Maximum length 32 characters

Processing Type

The type of processing associated with the module is entered in this field. Permissible codes for this field are as follows:

B Batch processed
 T Teleprocessed
 n User-assigned values 0 through 9

Field characteristics are as follows:

System name TYPEPRG
 Report name TYPE
 Maximum length 1 character

Module Identification

The module identification is entered in this field. The module is called by this name or cataloged under this name. Field characteristics are as follows:

System name MODID
 Report name MODULE ID
 Maximum length 10 characters

Program Overlay Indicator

This field is used to specify whether the module can be overlaid in processing. Permissible codes for this field are as follows:

Y Yes, overlay is permitted
 N No, overlay is not permitted
 n User-assigned values 0 through 9

Field characteristics are as follows:

System name OVLPRG
 Report name OVERLAID
 Maximum length 1 character

ATTRIBUTES Category

Attributes of the module being described are entered in this category. The set of attributes for modules includes definition of the language, type of code, overlays, and segmentation.

Programming Language

This field is used to identify the language in which the module was written. Permissible codes for this field are as follows:

A Assembler
 C COBOL
 F FORTRAN
 M MARK IV
 P PL/I

Segmented

This field is used to indicate whether the module is segmented. Permissible codes for this field are as follows:

Y Yes, the module is segmented
 N No, the module is not segmented
 n User-assigned values 0 through 9

Field characteristics are as follows:

System name SEGMENTD
 Report name SEGMENTED
 Maximum length 1 character

Call Type Code

This field indicates whether the module code is reentrant or recursive. Permissible codes for this field are as follows:

- E Code is reentrant
- R Code is recursive
- n User-assigned values 0 through 9

Field characteristics are as follows:

System name	CALLTYPE
Report name	CALL TYPE
Maximum length	1 character

STORAGE Category

The form in which the module is stored, where it is stored, and how much space it occupies can be specified in this category. Several storage lines can be entered to describe the various forms in which a module is stored.

Stored Module Form

The form of storage for the module is identified by a code in this field. Permissible codes for this field are as follows:

- L Load module
- O Object module
- P Partitioned dataset
- R Relocatable
- S Source
- C Core image

Field characteristics are as follows:

System name	STYPE
Report name	TYPE
Maximum length	1 character

Storage Location

The location of the module is entered in this field. This describes the place where the module is maintained. Field characteristics are as follows:

System name	PLACE
Report name	PLACE
Maximum length	20 characters

Storage Media

The storage media for the module is entered in this field. Cards, disk, or listing, for example, could be specified. Field characteristics are as follows:

System name	MEDIUM
Report name	MEDIUM
Maximum length	20 characters

Storage Size

This field is used to specify the size of the module to be stored. The unit of measure for the storage size is indicated by the next field described. Field characteristics are as follows:

System name	SIZE
Report name	STORED SIZE
Maximum length	7 digits

Storage Size Unit of Measure

The unit of measure for the storage size field is entered in this field. Permissible codes for this field are as follows:

- B Bytes or characters
- C Cylinders
- K K bytes or characters
- P Pages
- R Records
- T Tracks
- n User-assigned values 0 through 9

Field characteristics are as follows:

System name	UNITSIZE
Report name	Depends on the code selected
Maximum length	1 character

OPERATING Category

A module is a user of computer resources. The extent of the resources required is documented in this category. Only one set of field entries is required for this category.

Step Name

The module is identified by the NCL or CCL tag name (JCL step name) for the module. Field characteristics are as follows:

System name	STEPNAME
Report name	STEP NAME
Maximum length	44 characters

Region

The central memory space (region size) required for the module is entered in this field. The specified value represents thousands of characters (bytes); that is, a value

of 192 means 192 000 characters (bytes). Field characteristics are as follows:

System name	REGION
Report name	REGION
Maximum length	4 digits

Expected Elapsed Time

The expected elapsed time for the module is entered in this field. This is the wall clock time. The unit of measure applicable to this field is indicated by the next field described. Field characteristics are as follows:

System name	TELAPSED
Report name	ELAPSED TIME
Maximum length	3 digits

Expected Elapsed Time Unit of Measure

The unit of measure for the expected elapsed time field is entered in this field. Permissible codes for this field are as follows:

H	Hours
M	Minutes
S	Seconds
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	UELAPSED
Report name	Depends on the code selected
Maximum length	1 character

Expected CPU Time

The expected CPU time required for executing the module is entered in this field. The unit of measure for this field is specified by the next field described. Field characteristics are as follows:

System name	TIMECPU
Report name	CPU TIME
Maximum length	3 digits

Expected CPU Time Unit of Measure

The unit of measure for the expected CPU time field is entered in this field. Permissible codes for this field are as follows:

H	Hours
M	Minutes
S	Seconds
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	UNITCPU
Report name	Depends on the code selected
Maximum length	1 character

Procedure

The procedure of which this module is a part is entered in this field. This is the procedure name, NCL or CCL (JCL PROC name). Field characteristics are as follows:

System name	NAMEPROC
Report name	PROC NAME
Maximum length	8 characters

Sequence in Procedure

The sequence number of the module within the JCL PROC is entered in this field. Field characteristics are as follows:

System name	SEQPROC
Report name	PROC SEQUENCE
Maximum length	2 digits

Lines to Print

The estimated number of print lines to be produced by the module is entered in this field. Field characteristics are as follows:

System name	LINES
Report name	LINES PRINTED
Maximum length	7 digits

Punched Cards

This field is used to enter the estimated number of cards to be punched for the module. Field characteristics are as follows:

System name	PUNCH
Report name	CARDS PUNCHED
Maximum length	7 digits

SCHEDULING Category

The scheduling of a module is a critical part of the information processing function. Normal and deadline due dates and times are specified in this category.

Normal Start Time

The normal start time scheduled for the module is entered in this field. The time is specified as four digits representing a 24-hour clock. Field characteristics are as follows:

System name NBEGIN
 Report name NORMAL START
 Maximum length 4 digits

Normal Stop Time

The normal stop time scheduled for the module is entered in this field. This is also a four-digit 24-hour clock field. Field characteristics are as follows:

System name NEND
 Report name NORMAL STOP
 Maximum length 4 digits

Normal Day of Run

The day the module is normally run is entered in this field. The day can be indicated by a number or by another technique of abbreviation. Field characteristics are as follows:

System name NDAY
 Report name NORMAL DAY
 Maximum length 10 characters

Deadline Start Time

The last possible time of day when the module can be started is entered in this field. This is a four-digit 24-hour clock field. Field characteristics are as follows:

System name DBEGIN
 Report name DEADLINE START
 Maximum length 4 digits

Deadline Stop Time

This field indicates the deadline stop time for the module. This is a four-digit 24-hour clock field. Field characteristics are as follows:

System name DEND
 Report name DEADLINE STOP
 Maximum length 4 digits

Deadline Day

The last day of the week when the module can be run is entered in this field. Field characteristics are as follows:

System name DDAY
 Report name DEADLINE DAY
 Maximum length 10 characters

Cycle of Run

The scheduled frequency of the run for the module is entered in this field. Permissible codes for this field are as follows:

D Daily
 W Weekly
 M Monthly
 Q Quarterly
 S Semiannually
 A Annually
 R On request
 n User-assigned values 0 through 9

Field characteristics are as follows:

System name SCYCLE
 Report name CYCLE
 Maximum length 1 character

Preceding Program

The name of the module that immediately precedes the module being described is entered in this field. It need not be a Catalogue name. Field characteristics are as follows:

System name PRECPRG
 Report name PRECEDING PROGRAM
 Maximum length 32 characters

PASSING Category

The names of items that are passed by the module can be documented in this category. Each passed item is documented by specifying its name on a separate passing line.

Passing Name

The name of the data being passed by the module is entered in this field. This is a data name; it need not be a Catalogue name. Field characteristics are as follows:

System name PASSING
 Report name PASSING
 Maximum length 32 characters

RELATIONAL Category

A module can be related to other Data Catalogue entities. It is convenient to relate programs to one another and also to the files they access, the reports they produce, the systems in which they are found, and the users who are responsible for them.

Related Entry Name

The Catalogue name of the entry to which the module is related is entered in this field. The name must be one identified to Data Catalogue. This field is required to specify a relationship. Refer to table 3-2 for valid relational entity types. Field characteristics are as follows:

System name	CATNAME
Report name	CATALOGUE NAME
Maximum length	32 characters

Partial Use Indicator

This field specifies whether all or part of the named entry is related to the module. Additional relational lines can be used to indicate precisely which portions of the entry are related. Permissible codes for this field are as follows:

Y	Yes, partial use only
N	No, the complete entry is related
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	PARTIAL
Report name	PARTIAL
Maximum length	1 character

Input/Output Indicator

The type of processing relationship that the module has with the named entry is indicated by this field. This would normally be specified for a related file or data base. Permissible codes for this field are as follows:

I	Input only
O	Output only
B	Both input and output

Field characteristics are as follows:

System name	IOCODE
Report name	I/O
Maximum length	1 character

Access Type Code

The relationship between a module and a file can be further defined by specifying the access options in effect. Permissible codes for this field are as follows:

A	All types of access
C	Creates only
D	Deletes
R	Reads

U Updates

n User-assigned values 0 through 9

Field characteristics are as follows:

System name	ACCESS
Report name	ACCESS
Maximum length	1 character

Access Cycle

This field indicates the frequency of scheduled access when the related item is a file. Permissible codes for this field are as follows:

D	Daily
W	Weekly
M	Monthly
Q	Quarterly
S	Semiannually
A	Annually
R	On request
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	CYCLEPRG
Report name	CYCLE
Maximum length	1 character

Frequency of Access

The estimated frequency with which the module accesses the file is entered in this field. Field characteristics are as follows:

System name	FREQPRG
Report name	FREQ
Maximum length	5 digits

Disposition

The disposition of a file is specified in this field. For example, NEW,PASS or SHR could be entered. Field characteristics are as follows:

System name	DISP
Report name	DISP
Maximum length	20 characters

MANUAL TASK ENTITY

A manual task can be described to Data Catalogue. Activities such as reconciling controls, checking error messages, and report distribution are manual tasks.

In addition to the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER), the following categories can be used to describe a manual task:

RESPONSIBILITY
 NAMES
 SCHEDULING
 RELATIONAL

RESPONSIBILITY Category

The responsibility for a manual task can be documented in this category. The history of responsibility can be traced by adding a new responsibility line each time the responsibility is reassigned; the status of the previous responsibility line can then be changed from current to past.

Status

This field specifies whether the responsibility line is for a current or past responsibility. Permissible codes for this field are as follows:

C Current responsibility
 P Past responsibility
 n User-assigned values 0 through 9

Field characteristics are as follows:

System name	STATUS
Report name	STATUS
Maximum length	1 character

Function Code

The role of the responsible individual or unit is entered in this field. Permissible codes for this field are as follows:

D Developed the task
 C Changed the task
 T Tested the task
 n User-assigned values 0 through 9

Field characteristics are as follows:

System name	FUNCTION
Report name	FUNCTION
Maximum length	1 character

Department

The name of the department assigned the responsibility for the manual task is entered in this field. The name entered need not be defined to Data Catalogue. Field characteristics are as follows:

System name	DEPT
Report name	DEPARTMENT
Maximum length	32 characters

Person

The name of the individual responsible for the manual task is entered in this field. Field characteristics are as follows:

System name	PERSON
Report name	PERSON
Maximum length	32 characters

Phone

This field contains the telephone number of the department or person responsible for the manual task. Either the number and extension or the area code and number can be entered. Field characteristics are as follows:

System name	PHONE
Report name	PHONE
Maximum length	10 characters

Title

The title of the responsible individual is entered in this field. Field characteristics are as follows:

System name	TITLE
Report name	TITLE
Maximum length	15 characters

Mail Station

This field is used to document the mail station code of the responsible individual or unit. Field characteristics are as follows:

System name	MAIL
Report name	MAIL
Maximum length	5 characters

Date

The date at which the described responsibility took effect is entered in this field. Field characteristics are as follows:

System name	DATE
Report name	DATE
Maximum length	6 characters

NAMES Category

The name and identifying number of a manual task can be specified in this category.

Task Name

The name by which the manual task is generally known is entered in this field. Field characteristics are as follows:

System name	NAME
Report name	TASK NAME
Maximum length	32 characters

Task Identifier

The identifying number associated with the manual task is entered in this field. Field characteristics are as follows:

System name	ID
Report name	TASK ID
Maximum length	10 characters

SCHEDULING Category

Scheduling information associated with the manual task is specified in this category. One set of entries is used to describe a manual task.

Normal Start Time

The normal start time scheduled for the manual task is entered in this field. The time is specified as four digits representing a 24-hour clock. Field characteristics are as follows:

System name	NBEGIN
Report name	NORMAL START
Maximum length	4 digits

Normal Stop Time

The normal stop time scheduled for the manual task is entered in this field. This is also a four-digit 24-hour clock field. Field characteristics are as follows:

System name	NEND
Report name	NORMAL STOP
Maximum length	4 digits

Normal Schedule Day

The day the manual task is normally performed is entered in this field. The day can be indicated by a number or by another technique of abbreviation. Field characteristics are as follows:

System name	NDAY
Report name	NORMAL DAY
Maximum length	10 characters

Deadline Start Time

The last possible time of day by which the manual task can be started to meet its deadline due date is entered in this field. This is a four-digit 24-hour clock field. Field characteristics are as follows:

System name	DBEGIN
Report name	DEADLINE START
Maximum length	4 digits

Deadline Stop Time

This field indicates the last possible time of day by which the manual task must be finished. This is a four-digit 24-hour clock field. Field characteristics are as follows:

System name	DEND
Report name	DEADLINE STOP
Maximum length	4 digits

Deadline Day

The last day of the week when the manual task must be performed is entered in this field. Field characteristics are as follows:

System name	DDAY
Report name	DEADLINE DAY
Maximum length	10 characters

Cycle of Run

The frequency with which the manual task is scheduled is entered in this field. Permissible codes for this field are as follows:

D	Daily
W	Weekly
M	Monthly
Q	Quarterly
S	Semiannually
A	Annually
R	On request
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	SCYCLE
Report name	CYCLE
Maximum length	1 character

Preceding Step

The step preceding the manual task is identified in this field. The name entered in this field need not be a Catalogue name. Field characteristics are as follows:

System name	PRECSTEP
Report name	PRECEDING STEP
Maximum length	32 characters

RELATIONAL Category

A manual task can be related to many other entries including other tasks, users, forms, and reports. The name of the related entry and characteristics of the relationship are entered in this category.

Related Entity Name

The Catalogue name of the entry to which the manual task is related is entered in this field. The related entry must already exist in the Catalogue because cross-reference pointers between the task and the entry are created. This field is required to create a relationship. Refer to table 3-2 for valid relational entity types. Field characteristics are as follows:

System name	CATNAME
Report name	CATALOGUE NAME
Maximum length	32 characters

Partial Usage Code

This field indicates that the named entry and the manual task are not in a full usage relationship. For example, the manual task could use some of the data in a file, but not all of the data. Additional relational lines can be used to indicate precisely which portions of the entry are related to the task. Permissible codes for this field are as follows:

Y	Yes, partial use only
N	No, the complete entry is related
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	PARTIAL
Report name	PARTIAL
Maximum length	1 character

Task Input/Output Code

The type of access the manual task has with the related entry is entered in this field. Permissible codes for this field are as follows:

I	Input only
O	Output only
B	Both input and output
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	IOCODE
Report name	I/O
Maximum length	1 character

Related Task Cycle

The frequency of access the manual task has with the related entry is entered in this field. Permissible codes for this field are as follows:

D	Daily
W	Weekly
M	Monthly
Q	Quarterly
S	Semiannually
A	Annually
R	On request
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	CYCLE
Report name	CYCLE
Maximum length	1 character

SYSTEM ENTITY

A system can be thought of as containing processes that in turn access resources. A system can include procedure entities from manual or computerized sources. All of the previously defined entity types can be said to be subordinate to systems because they can be contained within systems.

In addition to the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER), the following categories can be used to describe a system:

RESPONSIBILITY
NAMES
RELATIONAL

RESPONSIBILITY Category

The responsibility for a system can be documented in this category. The history of responsibility can be traced by adding a new responsibility line each time the responsibility is reassigned; the status of the previous responsibility line can then be changed from current to past.

Status

This field specifies whether the responsibility line is for a current or past responsibility. Permissible codes for this field are as follows:

- C Current responsibility
- P Past responsibility
- n User-assigned values 0 through 9

Field characteristics are as follows:

System name STATUS
 Report name STATUS
 Maximum length 1 character

Function Code

The role of the responsible individual or unit is entered in this field. Permissible codes for this field are as follows:

- D Developed the system
- C Changed the system
- T Tested the system
- n User-assigned values 0 through 9

Field characteristics are as follows:

System name FUNCTION
 Report name FUNCTION
 Maximum length 1 character

Department

The name of the department assigned the responsibility for the system is entered in this field. The name entered need not be defined to Data Catalogue. Field characteristics are as follows:

System name DEPT
 Report name DEPARTMENT
 Maximum length 32 characters

Person

The name of the individual responsible for the system is entered in this field. Field characteristics are as follows:

System name PERSON
 Report name PERSON
 Maximum length 32 characters

Phone

This field contains the telephone number of the department or person responsible for the system. Either the number and extension or the area code and number can be entered. Field characteristics are as follows:

System name PHONE
 Report name PHONE

Maximum length 10 characters

Title

The title of the responsible individual is entered in this field. Field characteristics are as follows:

System name TITLE
 Report name TITLE
 Maximum length 15 characters

Mail Station

This field is used to document the mail station code of the responsible individual or unit. Field characteristics are as follows:

System name MAIL
 Report name MAIL
 Maximum length 5 characters

Date

The date at which the described responsibility took effect is entered in this field. Field characteristics are as follows:

System name DATE
 Report name DATE
 Maximum length 6 characters

NAMES Category

The name and an identifying number associated with the system can be specified in this category.

System Name

The name by which the system is generally known is entered in this field. This name is not necessarily the Catalogue name of the system. Field characteristics are as follows:

System name NAME
 Report name SYSTEM NAME
 Maximum length 32 characters

System Identifier

The identifying number by which the system is known is entered in this field. Field characteristics are as follows:

System name ID
 Report name SYSTEM ID
 Maximum length 10 characters

RELATIONAL Category

The relationships that occur between the system and other entries can be described in this category. A system can be related to such entity types as files, programs, and resources.

Related Entry Name

The Catalogue name of the entry to which the system is related is entered in this field. The entry must already exist in the Catalogue. This field is required to establish a relationship. Refer to table 3-2 for valid relational entity types. Field characteristics are as follows:

System name	CATNAME
Report name	CATALOGUE NAME
Maximum length	32 characters

Partial Usage Code

This field signifies that the related entry is not fully used by the system. Subsequent relational lines can be entered to tie specific used items to the system. Permissible codes for this field are as follows:

Y	Yes, partial use only
N	No, the complete entry is related
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	PARTIAL
Report name	PARTIAL
Maximum length	1 character

Related Job Name

The system can also be related to a job step of the entry. Field characteristics are as follows:

System name	JID
Report name	JOB NAME
Maximum length	8 characters

Related Job Title

The title or name by which the related job is generally known is entered in this field to fully identify the related job. This is not the Catalogue name of the related job. Field characteristics are as follows:

System name	JNAME
Report name	JOB TITLE
Maximum length	32 characters

USER ENTITY

The user is a significant factor in the information processing cycle. A user can be a department, section, control unit, an individual, or a division.

In addition to the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER), the following categories can be used to describe a user:

PERSONNEL
RELATIONAL

PERSONNEL Category

The name and location of the user can be specified in this category.

Section Name

The name of the management unit section is entered in this field. Field characteristics are as follows:

System name	SECTION
Report name	SECTION
Maximum length	32 characters

Function Code

The current function of the unit is entered in this field. Permissible codes for this field are as follows:

D	Development
C	Change
T	Testing
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	FUNCTION
Report name	FUNCTION
Maximum length	1 character

Person Responsible

The person responsible for the section is identified in this field. The name of the manager or supervisor is specified. Field characteristics are as follows:

System name	PERSON
Report name	PERSON
Maximum length	32 characters

Title

The title of the responsible individual is entered in this field. Field characteristics are as follows:

System name	TITLE
Report name	TITLE
Maximum length	15 characters

Phone

The telephone number of the responsible individual is entered in this field. The number and extension or the number and area code can be specified. Field characteristics are as follows:

System name	PHONE
Report name	PHONE
Maximum length	10 characters

Mail

The internal mail address of the management unit is entered in this field. Field characteristics are as follows:

System name	MAIL
Report name	MAIL
Maximum length	5 characters

RELATIONAL Category

The management unit can be related to other entries. This provides the ability to affix responsibility for various resources and activities by relating the responsible unit to the resource or activity.

Related Entry Name

The Catalogue name of the related entry is entered in this field. The name must already be defined to Data Catalogue to create valid cross-references. Refer to table 3-2 for valid user relationships. Field characteristics are as follows:

System name	CATNAME
Report name	CATALOGUE NAME
Maximum length	32 characters

Partial Usage Code

This field signifies that the unit is not fully responsible for the related resource or activity. Subsequent relational lines can be entered to tie the specific responsibilities to the unit. Permissible codes for this field are as follows:

Y	Yes, partial use only
N	No, the complete entry is related
n	User-assigned values 0 through 9

Field characteristics are as follows:

System name	PARTIAL
Report name	PARTIAL
Maximum length	1 character

Related Person

The person within the management unit who is responsible for the related resource or activity is specified in this field. Field characteristics are as follows:

System name	RPERSON
Report name	PERSON
Maximum length	32 characters

Data Catalogue is a system of computer programs, files, and support materials. The Update function is used to maintain the system. Maintenance of the system focuses on the activities of entering new data, changing data on file, deleting entries or portions of entries from the file, and providing audit and control facilities.

The Update function can be used in either a batch or interactive environment. The description in this section assumes a batch environment. Refer to section 10 for on-line processing requirements.

DATA CATALOGUE FILE MAINTENANCE

Once a system has been initialized, the maintenance process begins. Proper maintenance of the system is necessary to provide an effective system. Various maintenance tasks need to be performed. With the Data Catalogue system, these tasks are handled with economy of effort, with ease of specification, and under careful control through the Update function. Tasks that can be performed include the following:

- Add new entries.
- Add data to existing entries.
- Change data on file.
- Delete data from the file.
- Delete entries from the file.
- Diagnose and report on errors when they are detected.
- Audit and log transactions that are applied to the file via hard copy reports.

UPDATE REQUEST STRUCTURE

A series of statements comprise an Update request. Some statements form commands while others submit data values or set run time options. The Update function is a transaction-driven process.

Several types of statements can be entered for an Update request. Statements that form commands or set run time options are as follows:

- \$UPDATE**
Identifies the request as an Update request.
- OPTIONS**
Identifies options for the Update request.
- ADD**
Initiates a new entry.

- CHG**
Initiates a change for an entry.
- DEL**
Requests deletion of an entry.

The \$UPDATE statement and the OPTIONS statement are required for an Update request. The ADD, CHG, and DEL statements are used to formulate commands for operations to be performed by the Update function. At least one command must be entered; any number of commands can be entered in one Update request.

Two additional types of statements are used in an Update request. These statements, which are used in conjunction with the ADD and CHG statements, are as follows:

- Category Header**
Identifies a category of data to be processed.
- Category Line Number**
Identifies a line number within a category.

Figure 4-1 illustrates a typical deck setup for an Update request.

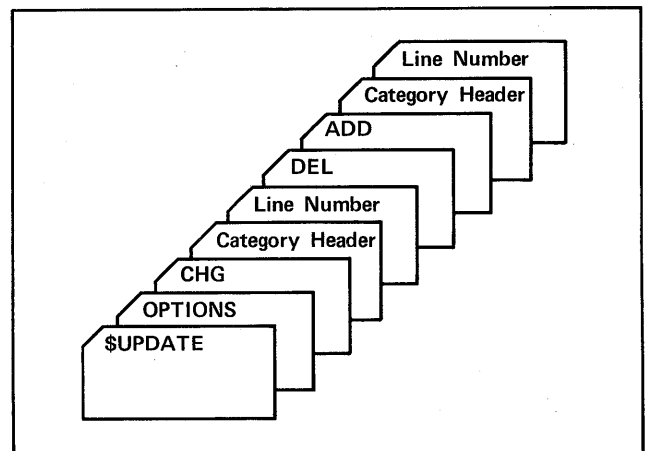


Figure 4-1. Update Request Deck Setup

\$UPDATE STATEMENT

The \$UPDATE statement is used to initiate an Update request. Only one \$UPDATE statement is required for an Update request. The format of the \$UPDATE statement is as follows:

\$UPDATE

This statement must be the first statement in an Update request.

OPTIONS STATEMENT

The OPTIONS statement for the Update function establishes options to be used when the Update request is executed. This must be the second statement in the Update request. Options that can be specified allow the user to control important facets of the updating process. The following options are available:

Validate the file edition by checking its revision number.

Update in an edit-only mode; the Catalogue is not updated on this run.

Produce or suppress a Program Revision Report.

Specify the character used to enclose alphanumeric literals.

Identify the user of the Update function for each changed entry.

Figure 4-2 illustrates the format of the OPTIONS statement. This statement is entered in free-form format.

```

OPTIONS {REV-NO= {NO-CHK} } [ ,QUOTE=c ]
        [ ,EDIT-ONLY= {YES} ] [ ,USER=uuu ]
    
```

Figure 4-2. OPTIONS Statement Format, Update Function

Phrases in the OPTIONS statement are defined as follows:

REV-NO=

Revision number; required. If the update is to match the latest file update revision number, the matching value must be supplied. One of the following values must be specified:

NO-CHK

The file revision number is not checked.

nnnn

The last revision number of the Catalogue must match the specified number; must be a numeric value.

EDIT-ONLY=

Edit-only mode for the Update request. Values that can be specified are as follows:

NO (default)

The Catalogue is updated by the request run.

YES

The Catalogue is not updated; the input transactions are edited for errors.

QUOTE=c

Character to enclose alphanumeric literals; default is double quotation mark ("").

USER=uuu

User identifier; default is no identifier. The three-character identifier is stored on the file for each changed entry.

The following rules are applicable to the OPTIONS statement:

Only one statement can be included in an Update request.

The REV-NO parameter must be specified in the statement.

The statement must be contained in one line; it cannot be continued.

Commas are used to separate phrases.

ADD STATEMENT

The ADD statement is a command to create a new entry in the Catalogue. The complete entry can be specified or only the Catalogue name. Any number of ADD statements can be included in an Update request. The format of the ADD statement is shown in figure 4-3.

```

ADD ent-type=catname
    
```

Figure 4-3. ADD Statement Format

Phrases in the ADD statement are defined as follows:

ent-type

Type of entity to be added to the Catalogue. Entity types that can be specified are as follows:

ELEMENT
GROUP
RECORD
FILE
DATASET
TOTAL
FORM
REPORT
EXTERNAL
TASK
MODULE
SYSTEM
USER

catname

Catalogue name to be assigned to the entry being added. It must not duplicate an existing Catalogue name.

CHG STATEMENT

The CHG statement is a command to access an existing entry in the Catalogue and proceed to change it. New

categories can be added or an entire category can be deleted. Individual category lines can be added or deleted. Fields in any category can also be added or deleted. Any number of CHG statements can be included in an Update request. The format of the CHG statement is shown in figure 4-4.

CHG ent-typ=catname

Figure 4-4. CHG Statement Format

Phrases in the CHG statement are defined as follows:

ent-type

Type of entity to be changed. Entity types that can be specified are as follows:

ELEMENT
GROUP
RECORD
FILE
DATASET
TOTAL
FORM
REPORT
EXTERNAL
TASK
MODULE
SYSTEM
USER

catname

Catalogue name of the entry to be changed.

The following rules are applicable to the CHG statement:

The specified Catalogue name must identify an existing Catalogue entry.

A Category Header statement must follow the CHG statement.

Unless the entire category is being deleted, at least one Category Line Number statement must follow the Category Header statement.

DEL STATEMENT

The DEL statement is a command to access an existing entry in the Catalogue and delete it entirely. When an entry is deleted, the Catalogue name of the entry is no longer recognized by Data Catalogue and none of the data entered under that name is accessible. Any number of DEL statements can be included in an Update request.

Upward and downward references to the entry being deleted can also be eliminated. If upward references are being deleted, the specified entry is eliminated as a component in higher level entries. If downward references are being deleted, all of the components of the entry are eliminated from the Catalogue. Both upward and downward references can be deleted by one statement. The format of the DEL statement is shown in figure 4-5.

**DEL ent-type=catname [WHEREUSED
USING
GLOBAL]**

Figure 4-5. DEL Statement Format

Phrases in the DEL statement are defined as follows:

ent-type

Type of entity to be deleted. Entity types that can be specified are as follows:

ELEMENT
GROUP
RECORD
FILE
DATASET
TOTAL
FORM
REPORT
EXTERNAL
TASK
MODULE
SYSTEM
USER

catname

Catalogue name of the entry to be deleted.

WHEREUSED

Deletes the specified entry from hierarchically higher entries.

USING

Deletes components of the specified entry.

GLOBAL

Deletes all upward and downward references to the specified entry.

The following rules are applicable to the DEL statement:

The specified Catalogue name must identify an existing Catalogue entry.

If WHEREUSED, USING, or GLOBAL is not specified, the entry is deleted only if it is not referenced by any other entry; otherwise, the deletion is not performed.

No additional statements are required for deleting entries. Any Category Header or Category Line Number statements after the DEL statement cannot access the deleted entry.

CATEGORY HEADER STATEMENT

A Category Header statement specifies the category of information to be processed when adding or changing an entry. If an entire category is being deleted from an entry, only the Category Header statement is required; otherwise, at least one Line Number statement must be

specified. A Category Header statement is used only after an ADD or CHG statement. The format of the Category Header statement is shown in figure 4-6.

```

cat-type [DELETE]
```

Figure 4-6. Category Header Statement Format

Phrases in the Category Header statement are defined as follows:

cat-type

System name for the category to be added or changed. Categories that can be specified are as follows:

CONTROL
CLASSIFICATION
DESCRIPTION
OTHER
ORIGIN
VALUES
ATTRIBUTES
ALIANSES
RESPONSIBILITY
NAMES
UNIT
SUPPLIERS
FLOW
DISTRIBUTION
SCHEDULING
LOCATION
OPERATING
PASSING
STORAGE
STRUCTURE
RELATIONAL
PERSONNEL
IOAREA
ENVIRONMENT

DELETE

Deletes the entire category.

The following rules are applicable to the Category Header statement:

This statement is used only following an ADD or CHG statement.

The DELETE option is valid only when processing a CHG statement.

The DELETE option cannot be used for the STRUCTURE, RELATIONAL, or CONTROL category; the DEL statement or a deletion by line numbers must be used for these categories.

At least one Category Line Number statement must follow the Category Header statement unless the entire category is being deleted (the DELETE option is specified).

CATEGORY LINE NUMBER STATEMENT

The lowest accessible item in an entry is a line number within a category. The line number is used to control the entry of data fields. Some data fields contain one character; others occupy several lines. When entering multiple-line fields (for example, in the DESCRIPTION category), line numbers are entered in ascending sequence. An interval between line numbers allows subsequent insertion of lines. Some categories have only one line number for the entire category. The format of the Category Line Number statement is shown in figure 4-7.

Phrases in the Category Line Number statement are defined as follows:

nnnn

Line number for the line to be added or changed; 1 through 9999.

sysname=value

System name of the field to be added or changed, and the value for the field.

\$DELETE

Deletes the specified line; if the THRU option is included, all lines from line nnnn to line tttt are deleted.

sn=\$DELETE

System name of the field to be deleted from the category.

The following rules are applicable to the Category Line Number statement:

When multiple statements are specified for a category, the line numbers must be in ascending sequence.

Line sequence numbers can ascend by any increment.

Any number of fields can be specified for a category line up to the line limit of 67 characters.

A category line can be continued to another line by entering a comma after a field specification and then entering a space on the next line before resuming field specifications.

A field specification (sys-name=value) cannot be continued from one line to the next.

```

nnnn [ $DELETE [THRU tttt]
      sysname={value $DELETE} [ ,sysname={value $DELETE} ] ... ]
```

Figure 4-7. Category Line Number Statement Format

Multiple line numbers for a single line category should be avoided because this wastes disk storage space; a full category line is stored for each line number specified, with null values for omitted system names.

Lines or fields can be deleted only when a CHG statement is being processed.

If a line to be deleted contains a mandatory field, the field is deleted and a diagnostic message is issued.

Any data field value that contains commas must be enclosed in quotation marks.

UPDATE ERROR PROCESSING

The Update function has a comprehensive editing facility. Each of the statements input to the Update function is checked for syntax and valid parameters. The contents of fields are also checked for valid information. Types of errors that can be diagnosed by the Update function are as follows:

Invalid code

Invalid format

Invalid length

Invalid system name

Specified entry not in the Catalogue

Attempt to add an entry already in the Catalogue

Attempt to delete or change an entry not in the Catalogue

Attempt to delete a line not defined for the category

Attempt to delete a category not specified in the entry

Attempt to delete a field not defined for the category

When an error is diagnosed, a message is output by the Update function. For some types of errors, the message indicates correct values for the field in error. Syntax errors are indicated as such at the beginning of the message. A code indicating the severity of the error is also output as part of the message. Refer to appendix B for more information on error processing and for a complete list of error messages.

AUDITING AND LOGGING TRANSACTIONS

When a transaction has successfully passed the editing process, it is applied to the Catalogue. Each such action is logged by the system as a means of providing an audit trail and a control on changes. The log of transactions is designed to show the transaction as it is entered into the system and the resulting Catalogue values after the transaction has been processed. In addition, the system shows the contents of the Catalogue field being changed before the change was applied, thus providing a complete audit trail of the updating process. Field values are not printed for a change or delete transaction that is not field oriented (for example, deleting an entire entry).

BACKUP AND RESTORE OF SYSTEM FILES

A final consideration in the maintenance of the Catalogue is the integrity of the files comprising the Catalogue. The files contain cross-reference data that is updated. If the system fails during an update, the cross-reference data could be incomplete or inaccurate. To protect against this contingency, a regular backup of the master files to secondary storage media should be performed. After a need has arisen, the files can be restored using the Utility function.

The backup and restore operations are performed when there is a need to consolidate the files or change the number of home blocks for the direct access files where the Catalogue resides. For a simple copy operation, the operating system COPY facilities can be used in place of the Data Catalogue backup and restore facilities of the Utility function.

During the backup and restore operations, a series of statistics are gathered and then displayed at the end of each process. The statistics show the count of entries by entity type and the number of physical records for each entity type. Refer to the Utility function description in section 9.

PRINTED STATISTICS

The Update Audit Report ends with a series of statistics that provide entry and transaction counts as follows:

TOTAL TX IMAGES READ = nnn

A count of transactions being applied; does not include comment lines.

TOTAL ENTRIES PROCESSED = nnn

A count of discrete entries that were processed; that is, every ADD, CHG, or DEL statement that identified an entry.

TOTAL ENTRIES ADDED = nnn

A count of new entries added to the file.

TOTAL ENTRIES CHANGED = nnn

A count of already existing entries that were updated.

TOTAL ENTRIES DELETED = nnn

A count of entries deleted from the Catalogue.

TOTAL SERIOUS ERRORS = nnn

A count of serious errors detected in the input.

TOTAL WARNINGS = nnn

A count of warning errors detected in the input.

TOTAL ent-type = nnn

A count of the number of entries processed for individual entity types.

Requests for information about the Catalogue can be made through the Query function. Query function input is in the form of a Query request; output is in the form of a response, which can be a display or a report. Typical information that can be requested by the Query function is as follows:

- Which record is used in both FILE-A and FILE-B?
- What are the aliases of UNIT-PRICE?
- Which programs process PRICE-PER-UNIT and PART-NUMBER?
- Which files in the Catalogue have no structure defined?

Questions can be based on the contents of fields in entries, the relationships between entries, hierarchical placement, or combinations of these characteristics. Queries can be simple or compound. All Catalogue entries can be scanned for the response to a query or the search can be limited to classes of entries.

The Query function can be used in either a batch or interactive environment with equal effectiveness. The description in this section assumes a batch environment. Refer to section 10 for on-line processing requirements.

QUERY FUNCTION CAPABILITIES

The syntax of the Query function is designed to facilitate communications between the user and Data Catalogue. The syntax allows queries to be expressed in terms that are very close to the English language.

The Query function is used as an information retrieval tool. Individuals seeking information about the contents of the Catalogue can have widely differing interests. A programmer might inspect a specific entry. An end user might want to find out about responsibilities. Forms control personnel might want to know about the usage of forms. Scheduling information could be requested by management.

The Query function can also be used as a data administration tool. The person responsible for the Catalogue (the data administrator) needs to be able to find out about the status of entries; for example, a list of entries that are missing vital categories of information can be requested. Data structures and entry relationships can be explored. Requests to determine which modules use a specific record or element, which elements are used by higher level entries, or which entries have no references or relationships defined can be initiated by the data administrator.

INFORMATION RETRIEVAL

Information retrieval functions are expressed through a simple search clause in a Query request statement. The search clause defines a class of entries by setting up one or more criteria that must be matched to satisfy the

information request. For example:

- Which elements have a length greater than 12?
- Which modules are scheduled for daily processing?
- How many forms have three parts?

In the first example, the search process examines all of the element entries and inspects the length attribute looking for those elements that have a specified length greater than 12 characters. The search criteria narrow the search for an entity type by selecting entries based on the contents of a selected attribute.

The search clause can also specify that entry selection be based on relationships. Simple relational queries could be as follows:

- Which elements are used by MODULE-X and MODULE-Y?
- How many files use GROUP-A and GROUP-B?
- Which report is used by PAYROLL-SYSTEM?

In these queries, the search proceeds from the known entry (MODULE-X, for example) to the unknown related entries (elements, for example). The unknown entries are related to the known entry either in a hierarchical chain or through a defined relationship. The search ranges across the defined relationships as well as up and down the defined hierarchies looking for entries that qualify.

The search clause in a Query request statement can combine aspects of the relationships and of attribute contents. For example:

- Which files with a block size of 2000 are used by SYSTEM-A?
- Which records with a minimum size of 500 use ELEMENT-A or ELEMENT-B?
- Which modules with a weekly cycle are used by PAYROLL-SYSTEM?

In these examples, the contents of an entry attribute must fulfill certain criteria (block size of 2000, for example), but the search is further narrowed to those entries meeting the relational or hierarchical requirements.

DATA ADMINISTRATION

The Query function can be used to meet many needs of data administration. In addition to information retrieval capabilities, the data administrator requires other capabilities provided by the Query function. The needs of the data administrator might include the following types of queries:

- Which elements do not have any attributes defined?

How many files have no defined components?

Which modules were just updated?

In these queries, the search is not based on the contents of the fields, but is based on the presence or absence of categories of information. With this capability, the data administrator can determine those entries that are incomplete. Another common need of the data administrator is the ability to locate entries that were entered or changed during the last update of the Catalogue.

Content searches can be combined with these other types of searches. This provides the data administrator with a truly powerful retrieval capability. For example:

Which records having no attributes defined are used by MODULE-X?

Which element with a length of 3 was just changed in a content search for an entry that had no attributes?

Which records used by SYSTEM-S are missing descriptions?

QUERY REQUEST STRUCTURE

A Query request consists of a series of statements that initiate a search of the Catalogue for entries meeting certain criteria. Qualifying entries are presented in a response; the kind of response is determined by the type of statement issued.

Statements applicable to a Query request are as follows:

\$QUERY

Identifies the request as a Query request.

TITLE

Provides an optional heading for responses.

COUNT

Determines the number of qualifying entries.

LIST

Lists the names of qualifying entries.

SHOW

Provides a detailed listing of qualifying entries.

A Query request is initiated by a \$QUERY statement. The TITLE statement is optional. At least one COUNT, LIST, or SHOW statement must be specified; any number of these statements can be included in the request. Comment lines can be specified to document the request being made.

A sequence number can be entered in columns 73 through 80 in any statement when the request is submitted on punched cards. Sequence numbers are not checked by Data Catalogue.

Figure 5-1 illustrates a typical deck setup for a Query request.

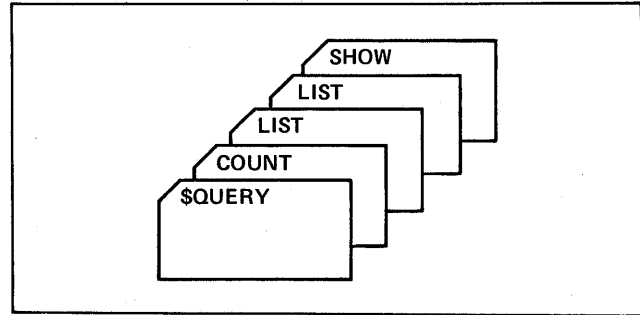


Figure 5-1. Query Request Deck Setup

\$QUERY STATEMENT

The \$QUERY statement is used to initiate a Query request. Only one \$QUERY statement can be specified in a Query request. The format of the \$QUERY statement is as follows:

\$QUERY

This statement must be the first statement in a Query request.

TITLE STATEMENT

The TITLE statement is used to place a title on the response produced as a result of the Query request. This is an optional statement. The TITLE statement is not applicable in the on-line mode of operation. The specified title is printed at the top of each page of the Query response report or display. The format of the TITLE statement for a Query request is as follows:

TITLE "titlename"

The single phrase in the TITLE statement is defined as follows:

"titlename"

Name of the title to be printed at the top of each page of the report or the first line of each display.

The following rules are applicable to the TITLE statement:

The statement cannot be continued onto a second line.

The specified title can be a maximum of 50 characters.

The title must be enclosed in quotation marks ("").

A quotation mark cannot be used within the title; an embedded quotation mark is considered to be the ending quotation mark and subsequent characters in the title are ignored.

When specified, the TITLE statement must immediately follow the \$QUERY statement.

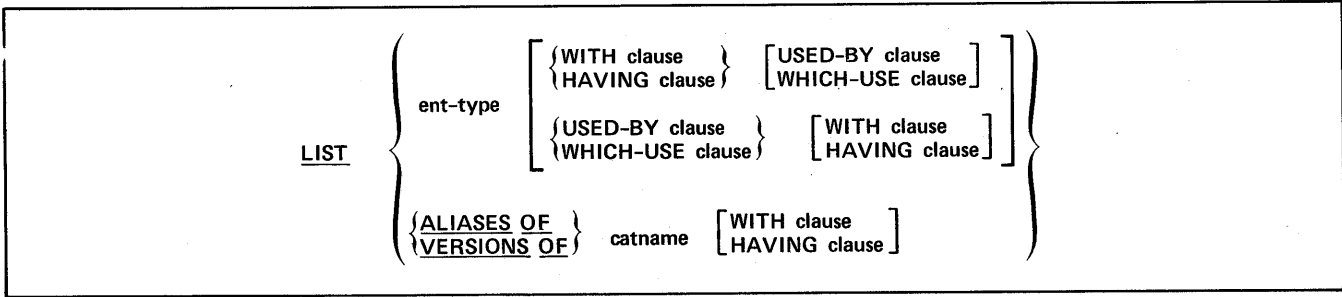


Figure 5-3. LIST Statement Format

MODULES
SYSTEMS
USERS
GDATA
GPROCEDURES
ENTRIES

When GDATA, GPROCEDURES, or ENTRIES is specified, qualified entries are all data entities, all procedure entities, or all entity types, respectively. The WITH, HAVING, USED-BY, and WHICH-USE clauses can be used to further limit the entries to be listed.

ALIASES OF or VERSIONS OF

Aliases or versions to be listed; all occurrences of entries that are aliases or versions of the specified Catalogue name are listed unless an optional search criteria clause is specified to limit the search.

The WITH, HAVING, USED-BY, and WHICH-USE clauses are detailed later in this section.

The following rules are applicable to the LIST statement:

The statement can be continued onto two additional lines.

Any number of LIST statements can be specified in a Query request.

LIST Statement Response

The response to a LIST statement is a line indicating the number of qualified entries encountered in the search followed by one line for each qualified entry. The response is output in the following format:

```
nnnnn ENTRIES QUALIFY
      catname      entity-type
      catname      entity-type
```

A line for a qualified entry contains the Catalogue name and the entity type.

SHOW STATEMENT

The SHOW statement is used to request a detailed listing of Catalogue entries that match the search criteria. The detailed listing can be limited to those entries of the specified type that meet certain criteria.

A detailed listing of entries that are aliases or versions of a specified Catalogue name can also be requested. The

SHOW statement initiates a search of the Catalogue for qualifying entries. The format of the SHOW statement is shown in figure 5-4.

SHOW Statement Phrases

Phrases in the SHOW statement are defined as follows:

CATEGORY cat-type

Optional limit on type and number of lines to be listed. The listing for a qualified entry is limited to a specified category or category group and optionally to a number of lines within the selected category. Categories and category groups that can be specified are as follows:

CONTROL
CLASSIFICATION
DESCRIPTION
OTHER
ORIGIN
VALUES
ATTRIBUTES
RESPONSIBILITY
NAMES
UNITS
SUPPLIERS
FLOW
DISTRIBUTION
SCHEDULING
LOCATION
OPERATING
PASSING
STORAGE
STRUCTURE
RELATIONAL
PERSONNEL
ADMINISTRATION
PEOPLE
CHARACTERISTICS
COMPONENTS
IOAREA
ENVIRONMENT
ALL

Within a specified category or category group, the number of lines to be listed can be limited by the following options:

FOR nnnn LINES

Specifies the maximum number of lines to be printed.

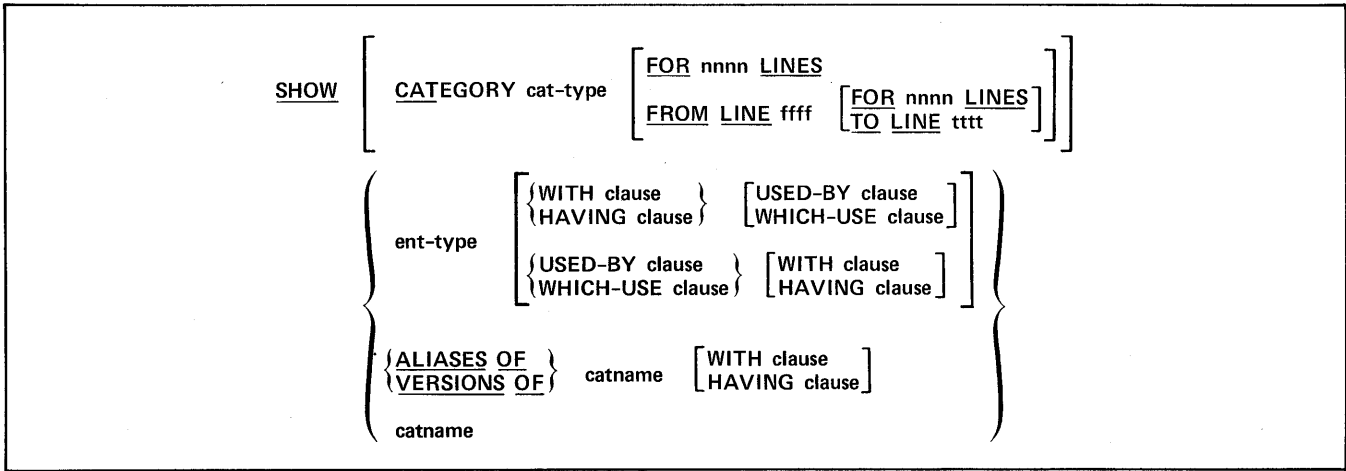


Figure 5-4. SHOW Statement Format

FROM LINE ffff FOR nnnn LINES

Specifies the first line number to be listed and the maximum number of total lines listed.

FROM LINE ffff TO LINE tttt

Specifies the first and last line number of lines to be listed.

ent-type

Type of entity to be listed in detail; all occurrences of the specified entity type are listed unless an optional search criteria clause is specified to limit the search. Entity types that can be specified are as follows:

- ELEMENTS
- GROUPS
- RECORDS
- FILES
- DATASET
- TOTAL
- FORMS
- REPORTS
- EXTERNALS
- TASKS
- MODULES
- SYSTEMS
- USERS
- GDATA
- GPROCEDURES
- ENTRIES

When GDATA, GPROCEDURES, or ENTRIES is specified, qualified entries are all data entities, all procedure entities, or all entity types, respectively. The WITH, HAVING, USED-BY, and WHICH-USE clauses can be used to further limit the entries to be listed.

ALIASES OF or VERSIONS OF

Aliases or versions to be listed in detail; all occurrences of entries that are aliases or versions of the specified Catalogue name are listed unless an optional search criteria clause is

specified to limit the search. The WITH and HAVING clauses can be used to further limit the listing.

The WITH, HAVING, USED-BY, and WHICH-USE clauses are detailed later in this section.

The following rules are applicable to the SHOW statement:

The statement can be continued onto two additional lines.

Any number of SHOW statements can be specified in a Query request.

When a category group is specified in the CATEGORY phrase, the DESCRIPTION category is always listed first. Table 5-1 lists the categories in each category group.

SHOW Statement Response

The response to a SHOW statement consists of three parts. The first part contains a line indicating the number of qualified entries encountered in the search. The second part contains the Catalogue name of each qualified entry. The third part contains a detailed listing for each qualified entry. The response is output in the format shown in figure 5-5.

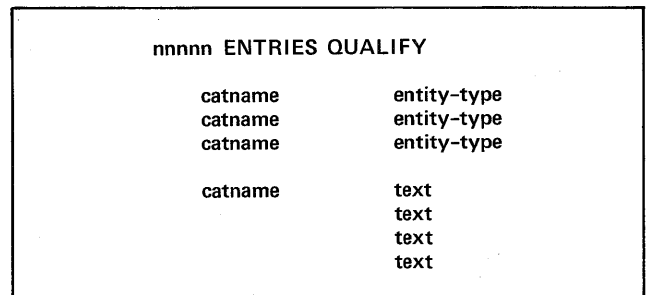


Figure 5-5. SHOW Statement Response Format

A line in the second part of the listing for a qualified entry contains the Catalogue name and the entity type. The third part of the listing consists of the Catalogue name for each qualified entry immediately followed by the detailed description, which is output in Catalogue Report format. (Refer to section 6 for the Catalogue Report format.)

TABLE 5-1. ENTITY TYPES AND CATEGORIES

Category or Group	ELE	GRO	REC	FIL	DAT	TOT	FOR	REP	EXT	TAS	MOD	SYS	USE	GDA	GPR	ENT
ADMINISTRATIVE																
CONTROL	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CLASSIFICATION	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
DESCRIPTION	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CHARACTERISTICS																
OTHER	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
ORIGIN	X															
VALUES	X															
ATTRIBUTES	X	X	X	X	X		X	X			X					
NAMES	X	X	X	X	X	X	X	X	X	X	X	X	X			
UNITS				X	X											
SUPPLIERS							X									
FLOW							X									
DISTRIBUTION								X								
SCHEDULING										X	X					
LOCATION											X	X				
OPERATING											X					
PASSING											X					
STORAGE											X					
IOAREA						X										
ENVIRONMENT					X											
COMPONENTS																
STRUCTURE		X	X	X	X	X										
RELATIONAL							X	X	X	X	X	X	X			
PEOPLE																
RESPONSIBILITY				X	X	X	X	X	X	X	X	X	X			
PERSONNEL																X

ELE = ELEMENTS
 GRO = GROUPS
 REC = RECORDS
 FIL = FILE
 DAT = DATASET
 TOT = TOTAL

FOR = FORMS
 REP = REPORTS
 EXT = EXTERNALS
 TAS = TASKS
 MOD = MODULES
 SYS = SYSTEMS

USE = USERS
 GDA = GDATA
 GPR = GPROCEDURES
 ENT = ENTRIES

SEARCH CLAUSES

The COUNT, LIST, and SHOW statements contain optional clauses that are used to further qualify the search for entries to be counted, listed, or shown. These clauses can be used separately or in conjunction with each other.

The optional search clauses are as follows:

WITH

Specifies field values for entries to qualify for the COUNT, LIST, or SHOW statement.

HAVING

Specifies status information for entries to qualify for the COUNT, LIST, or SHOW statement.

USED-BY

Specifies a higher-level entry or entity type for entries to qualify for the COUNT, LIST, or SHOW statement.

WHICH-USE

Specifies a lower-level entry or entity type for entries to qualify for the COUNT, LIST, or SHOW statement.

WITH Clause

The WITH clause is used to limit the search for qualified entries based on the contents of one, two, or three fields in the entry. The optional second and third fields are joined in logical relationships. The format of the WITH clause is shown in figure 5-6.

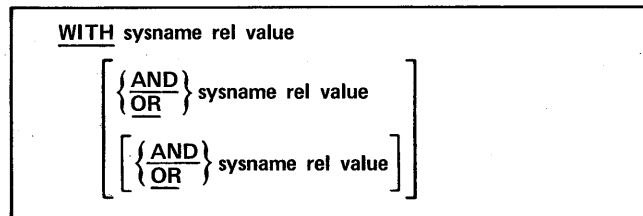


Figure 5-6. WITH Clause Format

Phrases in the WITH clause are defined as follows:

sysname

System name of a field in the entry. The system name can be abbreviated to the first three letters of the name. Refer to section 3 or appendix D for system names.

rel

Relationship of qualifying entry field value to value specified in the WITH clause. The relationship must be specified as one of the following:

- GT or > Greater than
- LT or < Less than
- EQ or = Equal to
- NE or N= Not equal to
- GE or >= Greater than or equal to
- LE or <= Less than or equal to
- CS Matching character string

value

Value to be compared with the specified field in the Catalogue entries. An unsigned numeric value or an alphanumeric literal can be specified. An alphanumeric literal need not be enclosed in quotation marks unless it includes spaces or special characters. If an alphanumeric literal is specified and the relationship (rel parameter) is not CS, only the first character is used in the comparison.

AND or OR

Logical connector for second or third field to be compared. AND requires a true condition for both fields. Or requires a true condition for at least one field. If AND and OR are both used in the WITH clause, the AND relationship is evaluated first.

The WITH clause can be used to search for entries that have a certain user-defined keyword specified in the CLASSIFICATION category of the entry. All keywords in the CLASSIFICATION category of an entry can be tested for the specified keyword, or only the keyword in a specific position in the classification line can be tested. For example, the search could be for all elements with ASSET as the third keyword.

Keywords can be tested for any logical relationship. Character string searches can be performed. A keyword search can be combined with search criteria specified for fields in other categories.

The system name for a keyword search is KWn. The n in the system name specifies the position of the desired keyword in the CLASSIFICATION category line. If n is omitted, the keyword can be in any position to qualify; otherwise, the keyword must be in the specified position to qualify.

The WITH clause can also be used to search for entries that have certain user-defined character combinations in the DESCRIPTION category of the entry. The free-form text in this category can be scanned for certain character combinations, or the entire category can be compared to a specified character combination. The system name for a DESCRIPTION category search is the category name DESCRIPTION or its abbreviated form DES.

When description lines are scanned for a character string, the entire character string in the DESCRIPTION category must be on the same line. Data Catalogue performs a line by line search for the character string.

HAVING Clause

The HAVING clause is used to limit the search for qualified entries based on certain status conditions. Conditions that can be checked are as follows:

Entries that are or are not referenced by other entries.

Entries that have specified category descriptions missing.

Entries that were or were not changed at a specified time (which can be a date, a revision number, or a number of times) or by a specified individual.

The format of the HAVING clause is shown in figure 5-7.

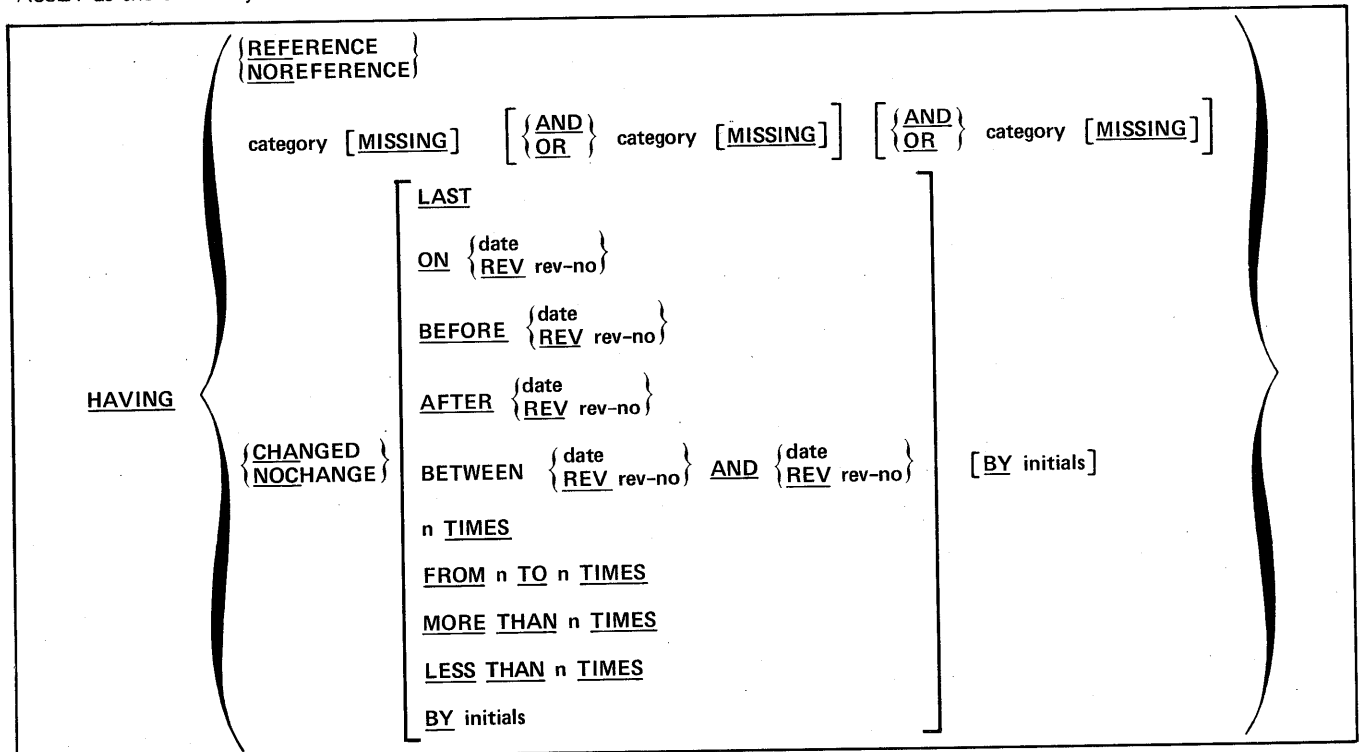


Figure 5-7. HAVING Clause Format

Phrases in the HAVING clause are defined as follows:

REFERENCE OR NOREFERENCE

Qualified entries are (REFERENCE) or are not (NOREFERENCE) referenced by other entries.

category

Data Catalogue category name or category group name. Refer to table 5-1 for category names and category group names valid for this phrase. If MISSING is specified, entries for which the category or group of categories are not defined qualify; otherwise, entries for which the category or group of categories are defined qualify. One or two additional phrases can be specified; these are connected in logical AND or logical OR relationships.

CHANGED or NOCHANGE

Search for entries that have or have not been changed within a specified limit. Criteria that can be specified are as follows:

LAST

Entries changed (or not changed) in the last update to the Catalogue.

ON date or ON REV rev-no

Entries changed (or not changed) on the specified date or revision number.

BEFORE

Entries changed (or not changed) before the specified date or revision number.

AFTER

Entries changed (or not changed) after the specified date or revision number.

BETWEEN

Entries changed (or not changed) at any time from the first date or revision number to the second date or revision number.

n TIMES

Entries changed (or not changed) the specified number of times.

FROM n TO n TIMES

Entries changed (or not changed) within and including the specified number of times.

MORE THAN n TIMES

Entries that have changed (or not changed) more than the specified number of times.

LESS THAN n TIMES

Entries that have changed (or not changed) less than the specified number of times.

BY initials

Entries changed (or not changed) by the individual with the specified initials. This phrase can also be used in conjunction with any of the preceding phrases to further limit the changed (or not changed) entries.

The following rules are applicable to the HAVING clause:

The REFERENCE or NOREFERENCE phrase cannot be used in conjunction with a USED-BY or WHICH-USE clause.

The first date or revision number in the BETWEEN or FROM phrase must be less than the second date or revision number in the phrase.

USED-BY Clause

The USED-BY clause specifies that the search is for entries referenced (or used by) higher-level entries. The search can be for a specific entry (identified by its Catalogue name) or for entries of a specific type. It can be used in conjunction with the WITH or HAVING phrase; the USED-BY clause can be applied either before or after the WITH or HAVING clause. The format of the USED-BY clause is shown in figure 5-8.

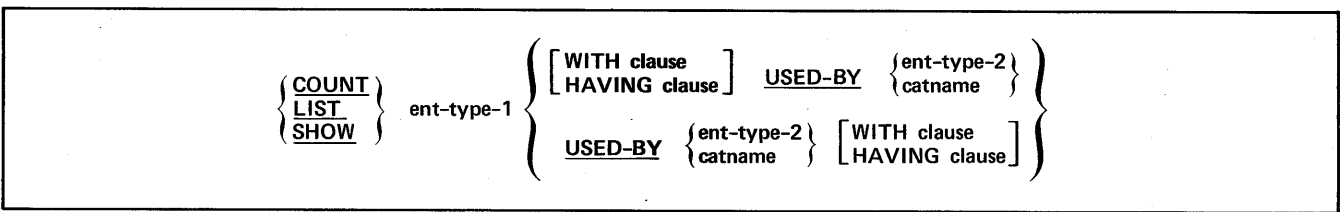


Figure 5-8. USED-BY Clause Format

Phrases in the USED-BY clause are defined as follows:

ent-type-1

Type of entry to qualify in the search. Refer to the COUNT, LIST, and SHOW statements for valid entity types.

WITH or HAVING clause

Limitation for entity type to qualify in the search. Refer to the WITH or HAVING clause description.

ent-type-2 or catname

Type of entity or Catalogue name of the higher-level entry to qualify in the search. Any entity type, except ELEMENTS, that is valid for the COUNT, LIST, or SHOW statement can be specified.

The following rules are applicable to the USED-BY clause:

The USED-BY clause cannot follow the HAVING REFERENCE or HAVING NOREFERENCE clause.

The entity type specified preceding the USED-BY clause must be at a lower level in the hierarchy than the entity type specified in the USED-BY clause; the only exception to this is that a different entity type at the same level can be specified. For example, a group can be used by a record or file but not by an element. Table 5-2 lists entity types in their hierarchical position.

TABLE 5-2. HIERARCHICAL POSITION OF ENTITY TYPES

ELEMENT			Lowest
GROUP			↓
RECORD			
FILE	DATASET		
TOTAL			
FORM	REPORT	EXTERNAL	
MODULE	TASK		
SYSTEM			
USER			Highest

WHICH-USE Clause

The WHICH-USE clause specifies that the search is for entries that reference (or use) lower-level entries. The search can be for a specific entry (identified by its Catalogue name) or for entries of a specific type. It can be used in conjunction with the WITH or HAVING phrase; the WHICH-USE clause can be applied either before or after the WITH or HAVING clause. The format of the WHICH-USE clause is shown in figure 5-9.

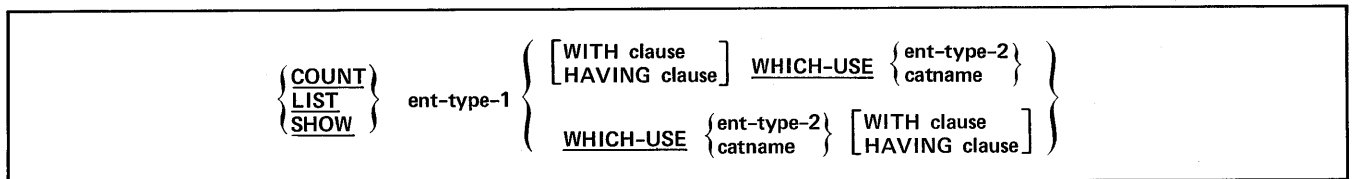


Figure 5-9. WHICH-USE Clause Format

Phrases in the WHICH-USE clause are defined as follows:

ent-type-1

Type of entity to qualify in the search. Refer to the COUNT, LIST, and SHOW statements for valid entity types.

WITH or HAVING clause

Limitation for entity type to qualify in the search. Refer to the WITH or HAVING clause description.

ent-type-2 or catname

Type of entity or Catalogue name of the lower-level entry to qualify in the search. Any entity type, except ELEMENTS, that is valid for the COUNT, LIST, or SHOW statement can be specified.

The following rules are applicable to the WHICH-USE clause:

The WHICH-USE clause cannot follow the HAVING REFERENCE or HAVING NOREFERENCE clause.

The entity type specified preceding the USED-BY clause must be at a higher level in the hierarchy than the entity type specified in the WHICH-USE clause; the only exception to this is that a different entity type at the same level can be specified. For example, a group can use an element or another group but not a record. Table 5-2 lists entity types in their hierarchical position.

Logical Connectors

Both the WITH and HAVING clauses can use the logical connectors AND and OR. The general format for using these logical connectors is shown in figure 5-10.

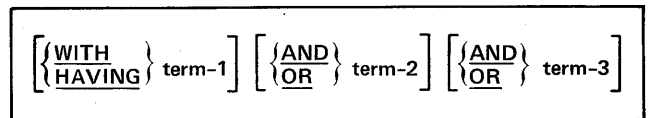


Figure 5-10. Logical Relationship Format

The following rules are applicable when using the logical connector AND:

The WITH or HAVING clause is evaluated as follows:

If term-1 is true and term-2 is true and only two terms are specified, the clause is true.

If term-1 is true, term-2 is true, and term-3 is true, the clause is true.

If any of the terms is not true, the clause is not true.

An AND relationship is always evaluated first; therefore, if three terms are connected as term-1 OR term-2 AND term-3, the term-2 AND term-3 relationship is evaluated first.

The following rules are applicable when using the logical connector OR:

The WITH or HAVING clause is evaluated as follows:

If term-1 or term-2 or term-3 is true, the clause is true.

If all of the specified terms are false, the clause is false.

If any term is true, the clause is true.

An OR relationship is always evaluated after any AND relationships. If three terms are connected by one AND and one OR, the terms connected by AND are evaluated first, and then the terms connected by OR are evaluated.

ent-type-2

Entity type that is the ending point for the search. Entity types at a higher level in the hierarchy than the specified entity type do not qualify for the search. Entity types that can be specified are as follows:

- GROUPS
- RECORDS
- FILES
- DATASET
- TOTAL
- FORMS
- REPORTS
- EXTERNALS
- TASKS
- MODULES
- SYSTEMS
- USERS

RANGE RETRIEVAL

The entries to be counted, listed, or shown can be limited in two additional ways. A range of entity types can be requested by specifying one level of entity type and a higher-level entity type. Those entity types within the specified range are counted, listed, or shown. A range can also be specified with the beginning and end of the range indicated by a particular Catalogue name or alphanumeric literal.

ENTITY TYPE RANGE

The COUNT, LIST, and SHOW statements can specify that the search for qualified entries includes a range of entity types rather than one specific entity type. For example, all entity types from elements to records can be searched for qualifying entries. The format for a statement searching a range of entity types is shown in figure 5-11.

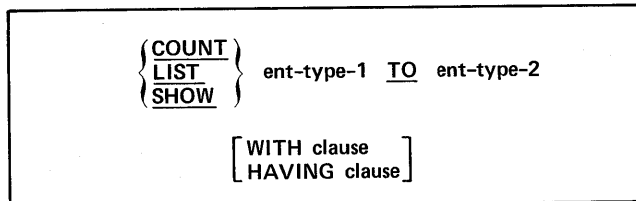


Figure 5-11. Entity Type Range Format

Phrases used to specify a range of entity types are defined as follows:

ent-type-1

Entity type that is the starting point for the search. Entity types at a lower level in the hierarchy than the specified entity type do not qualify for the search. Entity types that can be specified are as follows:

- ELEMENTS
- GROUPS
- RECORDS
- FILES
- DATASET
- TOTAL
- FORMS
- REPORTS
- EXTERNALS
- TASKS
- MODULES
- SYSTEMS

The following rules apply when an entity type range is specified:

The first entity type (ent-type-1) must be at a lower hierarchical level than the second entity type (ent-type-2); the only exception to this is that a different entity type at the same level can be specified. Refer to table 5-2 for the hierarchy of entity types.

The same entity type cannot be specified for ent-type-1 and ent-type-2.

If the WITH or HAVING clause is specified, it applies to both ent-type-1 and ent-type-2.

Only the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER) can be examined.

CATALOGUE NAME RANGE

The range of entries for a COUNT, LIST, or SHOW statement can be based on the Catalogue names of the entries. Actual Catalogue names or alphanumeric literals are used to identify the beginning and the end of the range. When this type of range is specified, the Catalogue name rather than the entity type determines whether an entry qualifies for counting or listing. The format for a statement searching a range of Catalogue names is shown in figure 5-12.

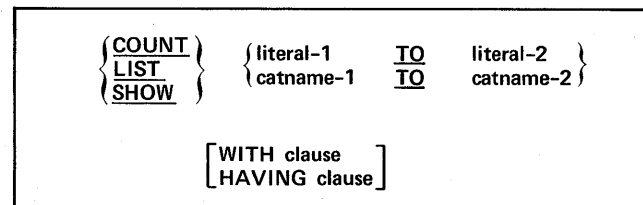


Figure 5-12. Catalogue Name Range Format

A Catalogue name range provides the means to examine sections of the Catalogue without knowing the names of the entries in that section. This is particularly useful when a set of entries share a common prefix. A facility exists to select sections of reports using a similar technique. A report can then be matched up to the entries used to produce it.

Phrases used to specify a range of Catalogue names are defined as follows:

literal-1 or catname-1

Alphanumeric literal or Catalogue name that defines the beginning of the search. Only Catalogue names that are equal to or greater than this value qualify for the search.

literal-2 or catname-2

Alphanumeric literal or Catalogue name that defines the end of the search. Catalogue names from the beginning of the range up to this value qualify for the search.

The following rules apply when a Catalogue name range is specified:

The first literal or Catalogue name must be less than the second literal or Catalogue name based on the collating sequence.

If the WITH or HAVING clause is specified, it applies to all Catalogue names within the range.

A WITH or HAVING clause can only be applied to the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER).

A variety of reports can be generated through Data Catalogue. Reports can be output in standard format, which produces pages in computer-width format, or in documentation format, which produces pages in 8-1/2 by 11 format. The Report function operates in batch mode only.

Data Catalogue reports used for documentation and data administration are as follows:

Catalogue Report

This report is the primary documentation tool in the Data Catalogue environment. It is a detailed presentation of all the data pertaining to a Catalogue entry. The report contains all the information needed to answer questions about an entry.

Usage Report

This report is a detailed presentation of an entry and of each higher-level entry that uses it. The entries are presented in Catalogue Report format.

Hierarchy Report

This report is a detailed presentation of an entry and of each lower-level entry used by it. This is the reverse of the Usage Report. The entries are presented in Catalogue Report format.

Data Catalogue reports used for analysis and data administration are as follows:

Index Report

This report produces an index for any standard Data Catalogue field except those that have free-form input. An index of Catalogue names (CATNAME), responsible departments (DEPT), or any other system name can be requested.

Relational Report

This report provides comprehensive cross-referencing of Data Catalogue fields. Cross-references can be obtained for any standard Data Catalogue system names except those that have free-form input. Both upward and downward relationships are included in the report.

Name Analysis Report

This report can be used to analyze standard Data Catalogue fields. It explodes compound names and values into their components and then groups identical components together.

Samples of these reports are shown in section 11.

REPORT REQUEST STRUCTURE

A Report request consists of a series of statements that initiate and control production of a Data Catalogue report. Statements applicable to a Report request are as follows:

\$REPORT

Identifies the request as a Report request.

TITLE

Provides an optional heading for the report.

OPTIONS

Defines printing options for the report.

SELECT

Controls selection of entries for the report.

OUTPUT

Limits the amount of output produced.

A Report request is initiated by a \$REPORT statement. The TITLE and OPTIONS statements are optional. Multiple SELECT and OUTPUT statements can be included in the request.

A sequence number can be entered in columns 73 through 80 in any statement. Sequence numbers are not checked by Data Catalogue.

Figure 6-1 illustrates the sequence of statements for a typical Report request.

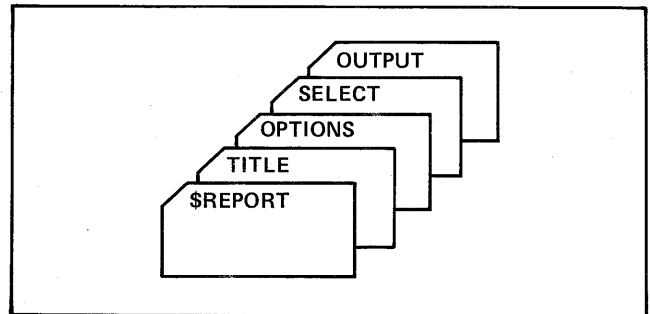


Figure 6-1. Report Request Deck Setup

\$REPORT STATEMENT

The \$REPORT statement is used to initiate a Report request. Only one \$REPORT statement can be specified

in a Report request. The format of the \$REPORT statement is shown in figure 6-2.

MODULES
SYSTEMS
USERS

RELATIONAL BY sysname

Selects the Relational Report; sysname is the system name for the Data Catalogue field for which the report is requested.

NAME-ANALYSIS

Selects the Name Analysis Report.

Additional phrases in the \$REPORT statement are defined as follows:

catname-1

Catalogue name of the entry for which the Usage or Hierarchy Report is to be prepared.

ent-type-1

Entity type for which the Usage or Hierarchy Report is to be prepared; the report includes all entries of the specified type. Entity types that can be specified are as follows:

- ELEMENTS
- GROUPS
- RECORDS
- FILES
- DATASET
- TOTAL
- FORMS
- REPORTS
- EXTERNALS
- TASKS

The report to be printed is selected by specifying one of the following options:

CATALOGUE

Selects the Catalogue Report.

USAGE OF

Selects the Usage Report.

HIERARCHY OF

Selects the Hierarchy Report.

INDEX

Selects the Index Report.

FOR catname-2 or FOR ent-type-2

Catalogue name or type of entity to which the Index, Relational, or Name Analysis Report is limited. Entity types that can be specified are the same as those listed for the Usage or Hierarchy Report.

USAGE

Include lower-level related entries in the Index, Relational, or Name Analysis Report; if omitted, higher-level related entries are included.

TITLE STATEMENT

The TITLE statement is used to enter an optional user title. The specified title is printed at the beginning of each page of the report. The format of the TITLE statement is as follows:

TITLE "user-title"

The following rules are applicable to the TITLE statement:

The specified title cannot exceed 50 characters.

The title must be enclosed in quotation marks.

A quotation mark cannot be used within the title; an embedded quotation mark is considered to be the ending quotation mark and subsequent characters in the title are ignored.

SELECT STATEMENT

The SELECT statement is used to specify the target entries to be shown in the report. The syntax for this statement is similar to the syntax for the Query function statements. The format of the SELECT statement for the Report function is shown in figure 6-3.

Phrases in the SELECT statement are defined as follows:

literal-1 TO literal-2

Alphanumeric literals identifying a range of Catalogue names to be selected for the report.

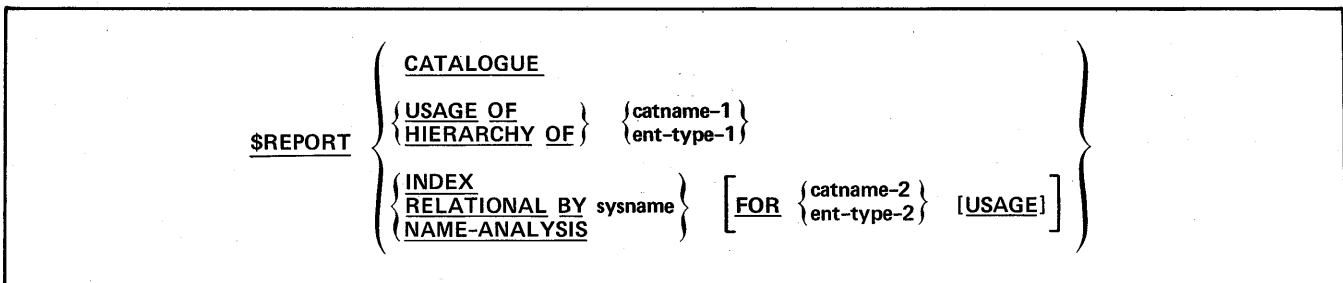


Figure 6-2. \$REPORT Statement Format

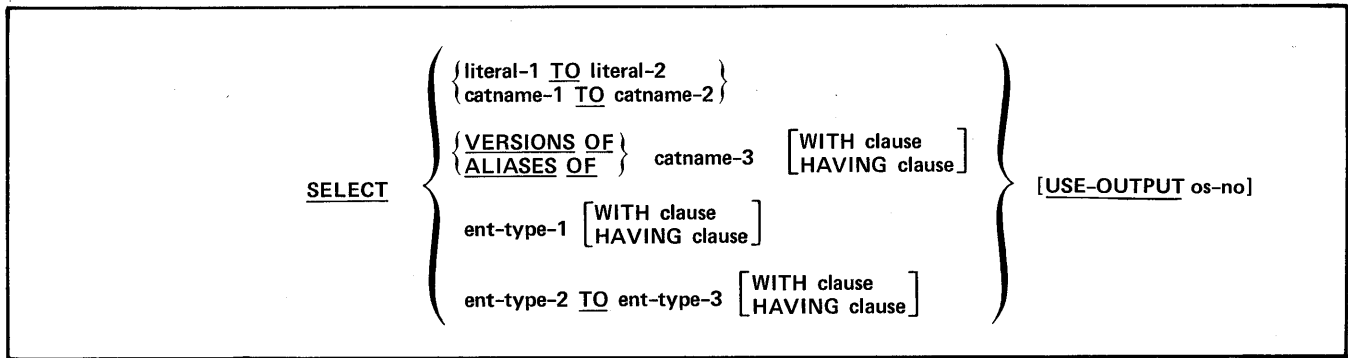


Figure 6-3. SELECT Statement Format, Report Function

catname-1 TO catname-2

Catalogue names identifying a range of Catalogue names to be selected for the report.

VERSIONS OF or ALIASES OF

Aliases or versions to be selected for the report. Entries that are aliases or versions of the specified Catalogue name are selected; the WITH or HAVING clause can be used to further limit the selected entries.

ent-type-1

Entity type to be selected for the report. All entries of the specified type are selected unless a WITH or HAVING clause is used to further limit the selected entries. Entity types that can be specified are as follows:

- ELEMENTS
- GROUPS
- RECORDS
- FILES
- DATASET
- TOTAL
- FORMS
- REPORTS
- EXTERNALS
- TASKS
- MODULES
- SYSTEMS
- USERS
- GDATA
- GPROCEDURES
- ENTRIES

When GDATA, GPROCEDURES, or ENTRIES is specified, entries selected are all data entities, all procedure entities, or all entity types, respectively.

ent-type-2 TO ent-type-3

Range of entity types to be selected for the report. All entries in the specified range are selected unless a WITH or HAVING clause is used to further limit the selected entries. Entity types that can be specified for ent-type-2 are as follows:

- ELEMENTS
- GROUPS
- RECORDS

- FILES
- DATASET
- TOTAL
- FORMS
- REPORTS
- EXTERNALS
- TASKS
- MODULES
- SYSTEMS

Entity types that can be specified for ent-type-3 are as follows:

- GROUPS
- RECORDS
- FILES
- DATASET
- TOTAL
- FORMS
- REPORTS
- EXTERNALS
- TASKS
- MODULES
- SYSTEMS
- USERS

USE-OUTPUT os-no

Number of the OUTPUT statement that applies to the SELECT statement.

The following rules are applicable to the SELECT statement:

If no SELECT statement is specified in the Report request, all entries that conform to the \$REPORT statement are selected for the report.

The first literal or Catalogue name must be alphabetically less than the second literal or Catalogue name.

If the SELECT statement specifies GDATA, GPROCEDURES, or ENTRIES, a WITH or HAVING clause can only reference the common categories (CONTROL, CLASSIFICATION, DESCRIPTION, and OTHER).

If ent-type-2 TO ent-type-3 is specified, ent-type-3 must be at a higher level in the hierarchy of entity types than ent-type-2 unless it is a different entity type at the same level.

If the \$REPORT statement includes the FOR catname phrase, the SELECT statement is processed against only those entries in the indicated hierarchy.

OUTPUT STATEMENT

The OUTPUT statement is used to limit the quantity of data that is output to a Catalogue, Usage, or Hierarchy Report. This statement is similar to the CATEGORY phrase in the SHOW statement for a Query request. The OUTPUT statement is not applicable to the Index Report, Relational Report, or Name Analysis Report. The format of the OUTPUT statement is shown in figure 6-4.

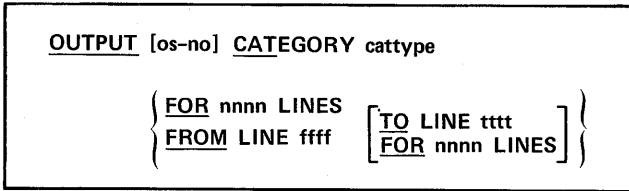


Figure 6-4. OUTPUT Statement Format

Phrases in the OUTPUT statement are defined as follows:

os-no

Output statement number associating the OUTPUT statement with a SELECT statement; 1 through 9 can be specified.

cattype

Category or category group name. Only data for the specified category or group of categories is printed. Refer to table 5-1 for category and category group names.

FOR nnnn LINES

Maximum number of lines to be printed for the specified category.

FROM LINE ffff TO LINE tttt

First and last line numbers of the lines to be printed for the specified category.

FROM LINE ffff FOR nnnn LINES

First line number and maximum number of lines to be printed for the specified category.

The following rules are applicable to the OUTPUT statement:

When an output statement number is specified, a SELECT statement with a matching USE-OUTPUT phrase statement number must be included in the Report request.

Output statement numbers must be defined in ascending sequence. The same output statement number can be assigned to more than one OUTPUT statement to provide more than one category of information.

SELECT statements can refer to the OUTPUT statements in any order.

A maximum of nine output statement numbers can be used in one Report request.

OPTIONS STATEMENT

The OPTIONS statement for a Report request sets options that control the particular report being processed. Options that can be specified depend on the report selected for printing. The OPTIONS statement is described in detail in the individual report descriptions.

DOCUMENTATION REPORTS

Three Data Catalogue reports are the source of basic documentation. The Catalogue Report provides a formatted presentation of all the data stored in an entry. The Usage and Hierarchy Reports give complete listings of an entry and other entries in its hierarchical structure.

CATALOGUE REPORT

The Catalogue Report is the basic documentation tool from Data Catalogue. It is a detailed presentation of all the data stored in the Catalogue for each entry. The format of the report is different for each entity type to accommodate differing data and reporting needs.

The Catalogue Report is the most complete account of any entry available. Other reports focus on relationships or uses; this report provides detail.

The request for a Catalogue Report can specify that the report be run as an archive or be run selectively. For example, the report can be for the whole Catalogue, for all the elements in the Catalogue, for a range of elements, or for a specific element. In addition, the amount of data included for each selected entry can be specified; all the data, specified categories of data, a maximum number of lines, or a range of specific line numbers can be shown in the report.

Other options that control the printing of the Catalogue Report can be specified in the OPTIONS statement. The format of the OPTIONS statement for a Catalogue Report is shown in figure 6-5.

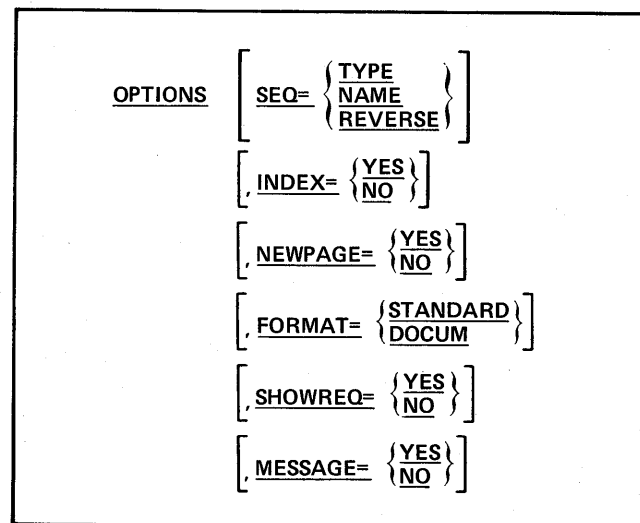


Figure 6-5. OPTIONS Statement Format, Catalogue Report

Options that can be specified in the OPTIONS statement are defined as follows:

SEQ=

Sequence of entries to be printed. Printing sequence is specified as one of the following:

TYPE (default)

The report begins with elements, then groups, and up through the hierarchy to users.

NAME

The report is in alphabetic sequence by Catalogue name.

REVERSE

The report is in descending hierarchical sequence.

INDEX=

Index preparation for entries appearing in the report. An index of Catalogue names in the report can be selected. Values for this option are as follows:

YES (default)

Prepare the index. In addition to Catalogue name, the index shows the entity type and the Catalogue Report page number for the entry. If SEQ=TYPE or SEQ=REVERSE is selected, a Catalogue Report table of contents is also prepared. This provides the first page number for selected entity types.

NO

Do not prepare the index.

NEWPAGE=

Print each selected entry on a new page. Values that can be specified are as follows:

YES (default)

The listing for each entry selected for the report begins on a new page.

NO

The listings for selected entries are printed continuously; page breaks do not occur for each entry. A new page begins only on page overflow.

FORMAT=

Standard or documentation format for printing. Values that can be specified are as follows:

STANDARD (default)

The report is printed using the standard computer page width for pages.

DOCUM

The report is printed using standard 8-1/2 by 11 page format.

SHOWREQ=

Print Report request as a leading page for the report. Values that can be specified are as follows:

YES (default)

The Report request that generated the report is printed before the report is printed.

NO

Printing of the Report request is suppressed.

MESSAGE=

Print installation-defined message at bottom of each page. Values that can be specified are as follows:

YES

The bottom of page message is printed on each page of the report.

NO (default)

No message appears at the bottom of each page of the report.

USAGE REPORT

The Usage Report begins with a detailed presentation of a selected Catalogue entry and then lists information related to each higher-level entry that uses it. Each entry is shown in Catalogue Report format.

This report is extremely useful in a detailed study of the usage of an entry. An entry such as an element could be selected. The Usage Report would then provide information about the element and about the groups, records, files, and other higher-level entries that use or are related to the element. The report could also be limited to one entity type to be shown.

Considerable flexibility is provided in selecting the information shown on the report. Output for the Usage Report can be limited to specified entity types, to specified categories of information, or to entries that meet specified criteria; SELECT statements are used for this type of limitation. OUTPUT statements can be used to limit the amount of information printed for selected categories. Other options, which control the printing of the Usage Report, can be specified by the OPTIONS statement. Figure 6-6 shows the format of the OPTIONS statement for the Usage Report.

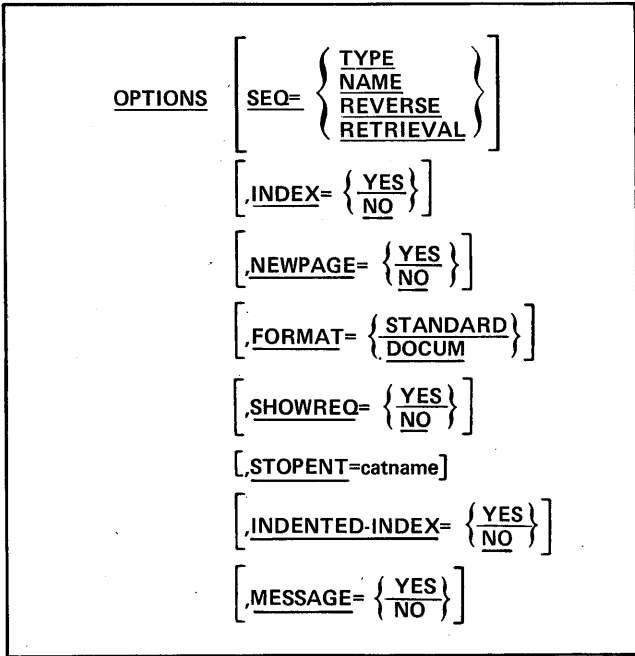


Figure 6-6. OPTIONS Statement Format, Usage Report

Options that can be specified in the OPTIONS statement are defined as follows:

SEQ=

Sequence of entries to be printed. Printing sequence is specified as one of the following:

TYPE (default)

The report begins with elements, then groups, and up through the hierarchy to users.

NAME

The report is in alphabetic sequence by Catalogue name.

REVERSE

The report is in descending hierarchical sequence.

RETRIEVAL

The report is in the order retrieved; that is, in where-used order.

INDEX=

Index preparation for entries appearing in the report. An index of Catalogue names in the report can be selected. Values for this option are as follows:

YES (default)

Prepare the index. In addition to Catalogue name, the index shows the

entity type and the Usage Report page number for the entry.

NO

Do not prepare the index.

NEWPAGE=

Print each selected entry on a new page. Values that can be specified are as follows:

YES (default)

The listing for each entry selected for the report begins on a new page.

NO

The listings for selected entries are printed continuously; page breaks do not occur for each entry. A new page begins only on page overflow.

FORMAT=

Standard or documentation format for printing. Values that can be specified are as follows:

STANDARD (default)

The report is printed using the standard computer page width for pages.

DOCUM

The report is printed using standard 8-1/2 by 11 page format.

SHOWREQ=

Print Report request as a leading page for the report. Values that can be specified are as follows:

YES

The Report request that generated the report is printed before the report is printed.

NO (default)

Printing of the Report request is suppressed.

STOPENT=catname

Catalogue name of the entry at which to stop the listing. Higher-level entries are not included in the report.

INDENTED-INDEX=

Indicates whether an indented index is to be produced. Values that can be specified are as follows:

YES

An indented index is generated.

NO (default)

An indented index is not generated.

MESSAGE=

Print installation-defined message at bottom of each page. Values that can be specified are as follows:

YES

The bottom of page message is printed on each page of the report.

NO (default)

No message appears at the bottom of each page of the report.

HIERARCHY REPORT

The Hierarchy Report begins with a detailed presentation of a Catalogue entry and then lists information related to each lower-level entry that is used by it. This report is a formatted listing of a hierarchy of entries beginning with the highest level entry in the hierarchy. The Hierarchy Report is the reverse of the Usage Report. Each entry is shown in Catalogue Report format.

This type of report has many uses. A programmer can produce program documentation by a Hierarchy Report that begins with a program entity. An analyst studying a system can obtain all information about that system from a Hierarchy Report beginning with the system. A project leader can isolate and review the complete activities of the project by requesting a report starting with the management group unit.

The information produced in a Hierarchy Report is determined by the SELECT, OUTPUT, and OPTIONS statements included in the Report request. The format of the OPTIONS statement for a Hierarchy Report is shown in figure 6-7.

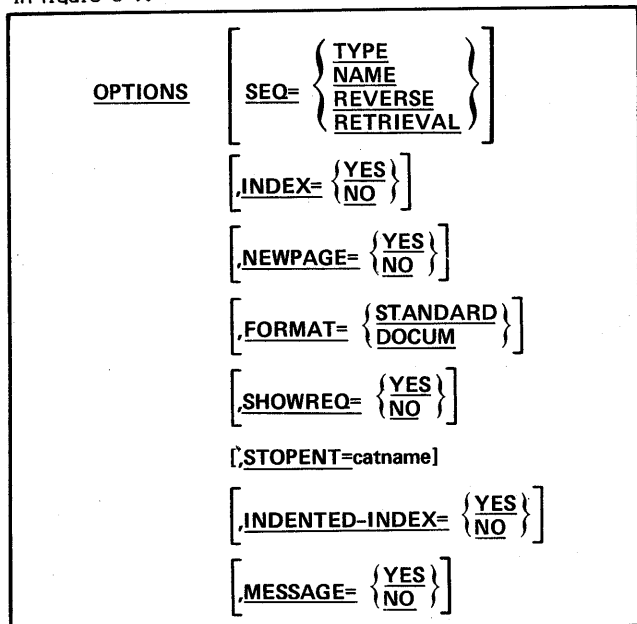


Figure 6-7. OPTIONS Statement Format, Hierarchy Report

Options that can be specified in the OPTIONS statement are defined as follows:

SEQ=

Sequence of entries to be printed. Printing sequence is specified as one of the following:

TYPE (default)

The report begins with elements, then groups, and up through the hierarchy to users.

NAME

The report is in alphabetic sequence by Catalogue name.

REVERSE

The report is in descending hierarchical sequence.

RETRIEVAL

The report is in the order retrieved; that is, down the hierarchy in sequence.

INDEX=

Index preparation for entries appearing in the report. An index of Catalogue names in the report can be selected. Values for this option are as follows:

YES (default)

Prepare the index. In addition to Catalogue name, the index shows the entity type and the Hierarchy Report page number for the entry.

NO

Do not prepare the index.

NEWPAGE=

Print each selected entry on a new page. Values that can be specified are as follows:

YES (default)

The listing for each entry selected for the report begins on a new page.

NO

The listings for selected entries are printed continuously; page breaks do not occur for each entry. A new page begins only on page overflow.

FORMAT=

Standard or documentation format for printing. Values that can be specified are as follows:

STANDARD (default)

The report is printed using the standard computer page width for pages.

DOCUM

The report is printed using standard 8-1/2 by 11 page format.

SHOWREQ=

Print Report request as a leading page for the report. Values that can be specified are as follows:

YES

The Report request that generated the report is printed before the report is printed.

NO (default)

Printing of the Report request is suppressed.

STOPENT=catname

Catalogue name of the entry at which to stop the listing. Lower-level entries are not included in the report.

INDENTED-INDEX=

Indicates whether an indented index is to be produced. Values that can be specified are as follows:

YES

An indented index is generated.

NO (default)

An indented index is not generated.

MESSAGE=

Print installation-defined message at bottom of each page. Values that can be specified are as follows:

YES

The bottom of page message is printed on each page of the report.

NO (default)

No message appears at the bottom of each page of the report.

ANALYSIS REPORTS

Three Data Catalogue reports are provided for data and resource analysis purposes. The Index Report produces an index based on a system name. The Relational Report is a listing of cross-referenced entries based on a system name. The Name Analysis Report lists compound Catalogue names according to the component parts of the name. A Report request for any of these reports does not include the OUTPUT statement.

INDEX REPORT

The Index Report is an important analytical tool. It is an index of Data Catalogue entries for the standard system

name of a field. The system name for any Data Catalogue field other than those having free-form input can be used as the basis for an Index Report.

The Index Report has a great number of uses. Indexes can be used to locate duplicate data names, analyze data name usage, and establish standard data names. A project analyzing the files in use at the installation could request an index of file names to provide a convenient starting point for the project.

Indexes are extremely valuable in performing system studies. An index by form type is useful in an evaluation of suppliers, or an index by central memory (region) size could be used in an analysis of hardware requirements.

The SELECT and OPTIONS statements are used to specify the output requirements for the Index Report. The full power of the SELECT statement is available to screen out items that do not match specified search criteria. The format of the OPTIONS statement for an Index Report is shown in figure 6-8.

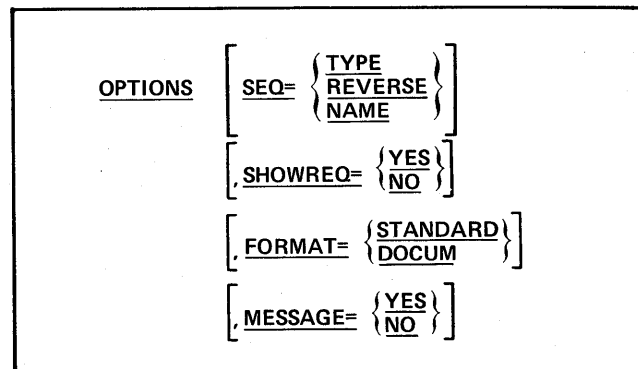


Figure 6-8. OPTIONS Statement Format, Index Report

Options that can be specified in the OPTIONS statement are defined as follows:

SEQ=

Sequence of entries to be printed. Printing sequence is specified as one of the following:

TYPE (default)

The report begins with elements, then groups, and up through the hierarchy to users.

NAME

The report is in alphabetic sequence by Catalogue name.

REVERSE

The report is in descending hierarchical sequence.

FORMAT=

Standard or documentation format for printing. Values that can be specified are as follows:

STANDARD (default)

The report is printed using the standard computer page width for pages.

DOCUM

The report is printed using standard 8-1/2 by 11 page format.

SHOWREQ=

Print Report request as a leading page for the report. Values that can be specified are as follows:

YES (default)

The Report request that generated the report is printed before the report is printed.

NO

Printing of the Report request is suppressed.

MESSAGE=

Print installation-defined message at bottom of each page. Values that can be specified are as follows:

YES

The bottom of page message is printed on each page of the report.

NO (default)

No message appears at the bottom of each page of the report.

RELATIONAL REPORT

The Relational Report is a valuable tool for providing summary documentation of relationships existing through the system files. Relationships can be user-defined (direct) or indirect. Direct relationships are specified through the RELATIONAL and STRUCTURE categories. Indirect relationships are established through direct relationships; for example, an element used in a record, which is in turn used by a module, is indirectly related to the module. The module could, but does not in most cases, relate directly to the element.

The Relational Report can provide rapid answers to important questions. The data administrator can review entries for their patterns of usage. An analyst can determine which reports, forms, and files use a particular field. Operations can learn which files, programs, and tasks are part of a system.

A Relational Report by Catalogue name shows all related entries including those related by alias or version fields. A Relational Report by any other field shows related entries and indicates where the particular field value was defined. A report by data name, for example, shows where each selected data name is defined and indicates which entries reference the defining field. Duplicate values are noted in the report.

A request for a Relational Report provides considerable flexibility. The \$REPORT statement can limit the output to entries related to a particular Catalogue name or a particular type of entity; it can further limit the output to higher or lower level entries in the hierarchy. The

SELECT statement capabilities can also be employed. The OPTIONS statement can be included in the Report request to specify the format of the output. The format of the OPTIONS statement for a Relational Report is shown in figure 6-9.

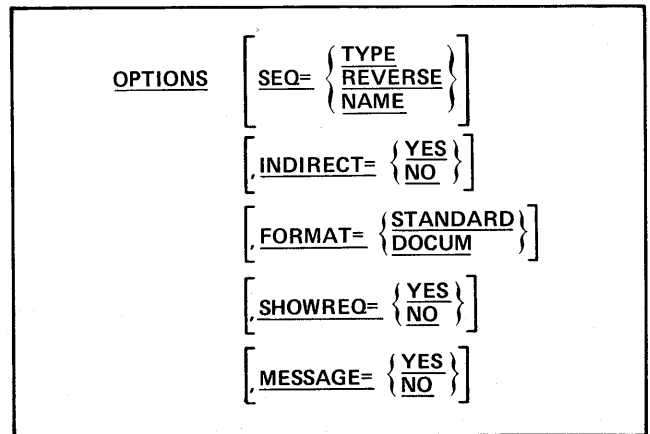


Figure 6-9. OPTIONS Statement Format, Relational Report

Options that can be specified in the OPTIONS statement are defined as follows:

SEQ=

Sequence of entries to be printed. Printing sequence is specified as one of the following:

TYPE (default)

The report begins with elements, then groups, and up through the hierarchy to users.

NAME

The report is in alphabetic sequence by Catalogue name.

REVERSE

The report is in descending hierarchical sequence.

INDIRECT=

Include or exclude indirectly related entries. Values that can be specified are as follows:

YES (default)

Indirect references are included.

NO

Indirect references are not included.

FORMAT=

Standard or documentation format for printing. Values that can be specified are as follows:

STANDARD (default)

The report is printed using the standard computer page width for pages.

DOCUM

The report is printed using standard 8-1/2 by 11 page format.

SHOWREQ=

Print Report request as a leading page for the report. Values that can be specified are as follows:

YES (default)

The Report request that generated the report is printed before the report is printed.

NO

Printing of the Report request is suppressed.

MESSAGE=

Print installation-defined message at bottom of each page. Values that can be specified are as follows:

YES

The bottom of page message is printed on each page of the report.

NO (default)

No message appears at the bottom of each page of the report.

NAME ANALYSIS REPORT

The Name Analysis Report is a listing of compound names that have a common name component. For example, the elements COUNT-INVENTORY and INVENTORY-COUNT would be listed for the components COUNT and INVENTORY. The Name Analysis Report explodes out a compound name and provides an alphabetic list of entries using the same component.

This report is an excellent means for detecting redundancies. It can be used to identify redundant elements, records, reports, forms, or other system entries. Redundant items can then be eliminated from the system.

A request for a Name Analysis Report provides considerable flexibility. The \$REPORT statement can limit the output to a particular Catalogue name or a particular type of entity; it can further limit the output to higher or lower level entries in the hierarchy. The SELECT statement capabilities can also be employed. The OPTIONS statement can be included in the Report request to specify the format of the output. The format of the OPTIONS statement for a Name Analysis Report is shown in figure 6-10.

Options that can be specified in the OPTIONS statement are defined as follows:

SEQ=

Sequence of entries to be printed. Printing sequence is specified as one of the following:

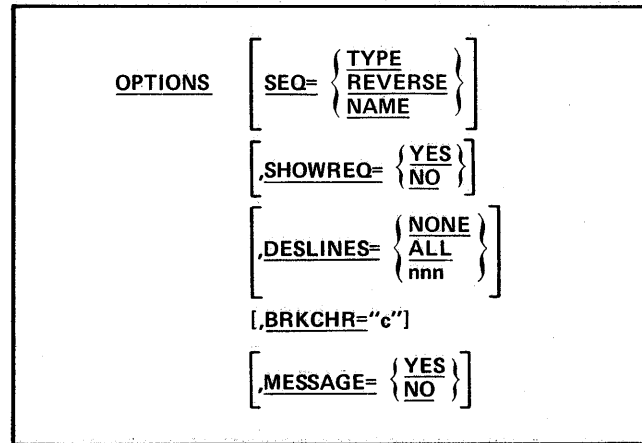


Figure 6-10. OPTIONS Statement Format, Name Analysis Report

TYPE (default)

The report begins with elements, then groups, and up through the hierarchy to users.

NAME

The report is in alphabetic sequence by Catalogue name.

REVERSE

The report is in descending hierarchical sequence.

SHOWREQ=

Print Report request as a leading page for the report. Values that can be specified are as follows:

YES (default)

The Report request that generated the report is printed before the report is printed.

NO

Printing of the Report request is suppressed.

DESLINES=

Number of DESCRIPTION category lines to be printed. Values that can be specified are as follows:

NONE (default)

No DESCRIPTION category lines are printed.

ALL

All DESCRIPTION category lines are printed.

nnn

The specified number of DESCRIPTION category lines are printed.

BRKCHR="c"

Character used to break a name into component parts; default is the hyphen (-).

MESSAGE=

Print installation-defined message at bottom of each page. Values that can be specified are as follows:

YES

The bottom of page message is printed on each page of the report.

NO (default)

No message appears at the bottom of each page of the report.

The Generate Function is provided to create data definitions in source statement format for use in COBOL and PL/I programs. Standard data definitions are generated from file and record entries in the Catalogue. The data definitions are created in the form of a sequential file. A listing of the generated data definitions can be requested in addition to or in place of the sequential file.

The TOTAL user can generate Data Base Definition Language (DBDL) statements from entries stored in the Catalogue. Naming practices can be standardized and more effective control over the data base environment can be maintained when this method is used. The source of the data base definition is the TOTAL data base entry that is stored in the Catalogue.

The use of the Generate function is quite similar for COBOL, PL/I, and TOTAL definitions. All three types of data definitions are derived from the structure of the data as described in the Catalogue. Considerations for each type of generation are presented after the statement descriptions.

The Generate function can be requested in a batch processing or on-line environment. This section assumes a batch environment; on-line processing requirements are described in section 10.

GENERATE CAPABILITIES

The Generate function enables programmers and analysts to request the automatic creation of data definitions for COBOL or PL/I programs or for TOTAL data bases. When the Generate function is used for TOTAL data bases, source statements for use with TOTAL are generated. The data base entry can be the source for a complete DBGEN with the exception of the OPTIONS, CONTROL-INTERVAL, and LOAD-LIMIT statements. With these exceptions, a complete DBDL is generated from the data base entry provided that the categories are complete.

Data Catalogue enables the programmer to generate language-dependent definitions for the same data item without redefining the item for the requirements of each language. This means that the same master definitions can be used completely independent of the language. The names used for a language definition are usually different; however, the structure and the working definitions are the same. This helps to unify and consolidate the development of systems.

A listing of the data definitions produced by the Generate function provides an additional benefit. As definitions evolve in an ongoing project, record and file structures gradually become complete. By using the list-only option of the Generate function, newly-developed structures can be scanned to determine if all the components are adequately defined. The listing indicates missing fields, thus enabling the programmer or analyst to complete the definition before generating the source statements.

GENERATE REQUEST STRUCTURE

A Generate request consists of a series of statements that initiate the request, specify run time options, and select the data structures or data bases to be used.

Three types of statements can be included in a Generate request:

\$GENERATE

Specifies a Generate request and the language to be used for generation.

OPTIONS

Selects run time options that govern the mode of generation.

SELECT

Identifies the data definitions to be generated.

The \$GENERATE statement initiates a Generate request for the specified language. The OPTIONS statement specifies those options that are to be used for the Generate request; this statement is optional. One or more SELECT statements must be included to indicate the Catalogue entries to be used in the generation of data definitions.

A sequence number can be entered in columns 73 through 80 in any statement when the request is submitted on punched cards. Sequence numbers are not checked by Data Catalogue.

Figure 7-1 shows the deck structure for a typical Generate request.

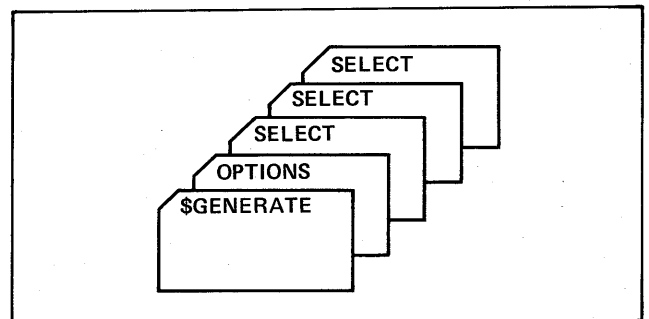


Figure 7-1. Generate Request Deck Setup

\$GENERATE STATEMENT

The \$GENERATE statement is required to initiate a Generate request. Only one \$GENERATE statement is specified for each request. The format of this statement is shown in figure 7-2.

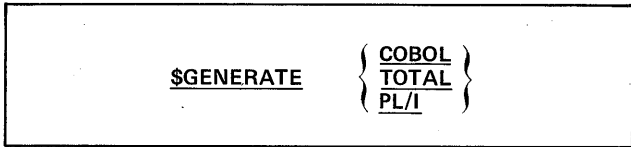


Figure 7-2. \$GENERATE Statement Format

The \$GENERATE statement has only one phrase, which is defined as follows:

COBOL

The data definitions are generated in the format for a COBOL program.

PL/I

The data definitions are generated in the format for a PL/I program.

TOTAL

The data definitions are generated in the format applicable to TOTAL.

The following rules are applicable to the \$GENERATE statement:

Only one \$GENERATE statement can be specified in one Generate request.

The \$GENERATE statement must be contained on one line.

Multiple DBDLs can be generated in one request.

OPTIONS STATEMENT

Run time options for a Generate request are specified by the OPTIONS statement. The statement is optional; only one can be specified for a request. Nine options are available; the default is effective for any option not specified. The format of the OPTIONS statement is shown in figure 7-3.

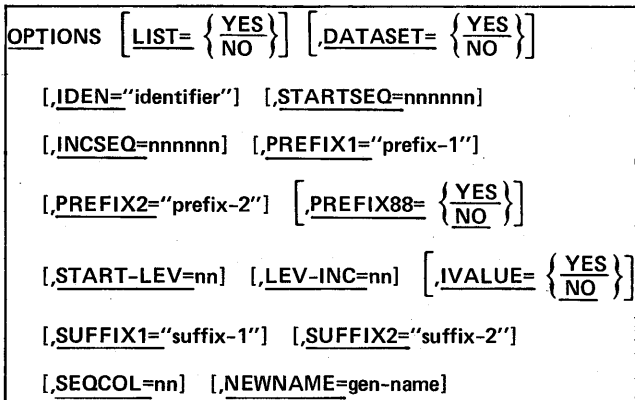


Figure 7-3. OPTIONS Statement Format, Generate Function

Phrases in the OPTIONS statement for the Generate function are defined as follows:

LIST=

Prints a listing of the generated data definitions and statements. Values that can be specified are as follows:

YES (default)

The listing is printed.

NO

The listing is not printed.

DATASET=

Produces a sequential file (PNCHFIL) for the generated data definitions and statements. Values that can be specified are as follows:

YES (default)

The sequential file is produced.

NO

The sequential file is not produced.

IDEN="identifier"

Inserts specified identifier in columns 73 through 80 of the sequential file records; not applicable to TOTAL. The identifier can have a value of up to eight characters. It is left-justified in columns 73 through 80. If an identifier is not specified, blanks are placed in columns 73 through 80.

STARTSEQ=nnnnnn

Inserts sequence numbers in columns 1 through 6 of the output beginning with the specified number. If a number is not specified, sequence numbers begin with 000000.

INCSEQ=nnnnnn

Increments sequence numbers by the specified value. This value is added to the current sequence number for each new definition or statement. If an increment value is not specified, Data Catalogue uses an increment value of 000010.

PREFIX1="prefix-1"

Adds a prefix to the beginning of names in generated data definitions; not applicable to TOTAL. The prefix can be up to nine characters in length and must be enclosed in quotation marks. To include a hyphen between the prefix and the generated name, the hyphen must be specified as part of the prefix.

PREFIX2="prefix-2"

Produces a second set of generated data definitions with a different prefix added to the beginning of names; not applicable to TOTAL. PREFIX2 cannot be specified if PREFIX1 is not specified. If a second prefix is not specified, only one set of definitions is generated.

PREFIX88=

Generates prefixes for COBOL level 88 data definitions. Values that can be specified are as follows:

YES (default)

Prefixes are generated for COBOL level 88 data definitions.

NO

Prefixes are not generated for COBOL level 88 data definitions.

START-LEV=nn

Starts COBOL level numbers with the specified value. If a level number is not specified, entries start with level 01.

LEV-INC=nn

Increments COBOL level numbers in the hierarchy by the specified value. If an increment value is not specified, level numbers are incremented by the value 02.

IVALUE=

Generates initial values; not applicable to TOTAL. Values that can be specified are as follows:

YES (default)

Initial values are generated.

NO

Initial values are not generated.

SUFFIX1="suffix-1"

Adds a suffix to the end of names in generated data definitions; not applicable to TOTAL. The suffix can be up to nine characters in length and must be enclosed in quotation marks. To include a hyphen between the generated name and the suffix, the hyphen must be specified as part of the suffix.

SUFFIX2="suffix-2"

Produces a second set of generated definitions with a different suffix added to the end of names; not applicable to TOTAL. SUFFIX2 cannot be specified if SUFFIX1 is not specified. If a second suffix is not specified, only one set of definitions is generated.

SEQCOL=nn

Begins sequence numbers in the specified column; not applicable to TOTAL. If a column number is not specified, sequence numbers begin in column 1 for COBOL and column 73 for PL/I.

NEWNAME=gen-name

Replaces the generated DBDL name with the specified name; not applicable to COBOL and

PL/I. A maximum of eight characters can be specified for the name. If a new name is not specified, the name in the data base entry is used.

The OPTIONS statement phrases and default values are summarized in table 7-1.

The following rules are applicable to the OPTIONS statement:

Selected options must be separated by commas.

The statement can be continued to three additional lines.

Both the prefix and suffix options cannot be selected in the same request.

PREFIX2 or SUFFIX2 cannot be selected unless PREFIX1 or SUFFIX1, respectively, is selected.

SELECT STATEMENT

The SELECT statement for the Generate function identifies a Catalogue entry that is to be used as the source for data definition generation. At least one SELECT statement must be included in a Generate request; multiple statements can be specified. The SELECT statement can identify the following:

File entry

Record entry

TOTAL data base entry

Procedure entry that uses one or more data bases

The format of the Generate function SELECT statement is shown in figure 7-4.

```
SELECT catname [DBDL]
```

Figure 7-4. SELECT Statement Format, Generate Function

Phrases in the SELECT statement are defined as follows:

catname

Specifies the Catalogue name of a file, record, or data base entry, or a procedure entry using at least one data base.

DBDL

Instructs Data Catalogue to inspect the structure lines of the specified entry and to generate all DBDLs encountered in the structure lines; applicable only to procedure entries.

The following rules are applicable to the SELECT statement:

Only one entry can be specified; multiple data structures or DBDLs are specified by multiple SELECT statements.

TABLE 7-1. SUMMARY OF OPTIONS AND DEFAULTS

Option	COBOL	PL/I	TOTAL	Default
LIST	X	X	X	A list of generated definitions is produced.
DATASET	X	X	X	A sequential file of generated definitions is produced.
IDEN	X	X		Spaces are inserted in columns 73 through 80 of output records.
STARTSEQ	X	X	X	Sequence numbers for columns 1 through 6 begin with zero.
INCSEQ	X	X	X	Sequence numbers are increased by 10.
PREFIX1	X	X		No prefix is added to generated data names or identifiers.
PREFIX2	X	X		No second set of definitions is generated.
PREFIX88	X			Prefix is added to level 88 entries.
START-LEV	X			Level number 01 is assigned to records.
LEV-INC	X			Level numbers are incremented by two.
IVALUE	X	X		Initial values are generated.
SUFFIX1	X	X		No suffix is added to generated data names or identifiers.
SUFFIX2	X	X		No second set of definitions is generated.
SEQCOL	X	X		Sequence numbers begin in column 1 for COBOL or column 73 for PL/I.
NEWNAME			X	Name defined in the data base entity is used.

The specified Catalogue name must identify an existing entry in the Catalogue.

When the DBDL option is specified for a data base entry, a DBDL is generated for that data base. If it is specified for a procedure entry, a DBDL is generated for each data base referenced in the structure. A Generate request can include both types of SELECT statements.

The specified Catalogue name can identify an entry that is an alias or version of another entry.

CONSIDERATIONS FOR GENERATION

All data definitions created through the Generate function are derived from the structure of the data as described in the Catalogue. Before a Generate request is initiated, various language-dependent factors should be considered.

COBOL CONSIDERATIONS

When the Generate function is used to create Data Division entries for a COBOL program, several

considerations should be reviewed. These considerations are discussed in the following paragraphs.

Basis for Generation

The Generate function uses the defined structure lines to determine which entries must be retrieved and the order in which they are retrieved. Structure lines are stored at the file, record, and group level. If a selected entry has no defined structure lines, no Data Division entries can be generated for that item. Similarly, if a file or record entry references a lower-level entry without any structure lines, the generation process cannot go beyond that entry. This condition is noted and the generation process continues with the next line in the structure.

As a structure line is followed through to the elementary level, the picture, format, length, and other attributes are retrieved from the Catalogue. The relative hierarchical position of each entry within the data structure being generated is maintained by the Generate function.

Records are normally assigned level 01; groups and elements within the record are assigned level numbers

consistent with their relative hierarchical positions. The first level number following the record level is normally level 03, which is in turn followed by level 05, and so on down the hierarchical line. The initial (record) level number and the increment value can be overridden by the OPTIONS statement.

Data Name Selection

The data name assigned to a generated data item is retrieved from the COBOL data name field in the NAMES category for the file, record, group, or element entry. If the COBOL data name field does not contain a name, the Catalogue name for the entry is assigned to the generated data item.

If the PREFIX1 option is selected by the OPTIONS statement, the specified prefix is inserted in front of the name retrieved from the COBOL data name field before completing the generation process. The entire prefix, which can consist of up to nine characters, is placed in the high order positions of the data name. If the resultant data name exceeds 32 characters, the low order characters are truncated.

If the PREFIX2 option is selected by the OPTIONS statement, a second set of definitions is generated. The data names for the second set of definitions are generated in the same manner as the first set, except that the PREFIX2 characters are inserted instead of the PREFIX1 characters.

The SUFFIX1 and SUFFIX2 options can be selected by the OPTIONS statement rather than the prefix options. Data names are generated in the same manner, except that the suffixes are appended to the end of the data names.

PICTURE Clause

The PICTURE clause for a data definition is generated from the PICTURE clause field in the element or group entry. Normally, the PICTURE clause field is defined at the element level and the specified picture is used in the generation process. When the PICTURE clause field is specified for a group entry, the picture is cascaded down to the elements within the group overriding any pictures specified for elements.

When no PICTURE clause is specified for a data definition being generated, Data Catalogue determines the picture from the length and format fields. For example, Data Catalogue generates the picture X(5) for an element with a defined length of five and format of character. The defaults for undefined length and format fields are one (1) and character (X), respectively. Group level length and format fields override element level specifications.

Filler Items

Any file, record, or group can have undefined space in its structure. This type of space is identified by the word FILLER in the Catalogue name field of the structure line. The length of the filler item is also defined by the filler length field in the structure line. The location of the filler item is indicated by the position of the structure line within the hierarchy of the file, record, or group. The Generate function creates a Data Division FILLER definition at the appropriate hierarchical level and assigns it a PICTURE clause of X(n), where n is the specified length for the filler item.

Other Entity Fields

Other entity fields that should be considered before using the Generate function are described in the following paragraphs.

Level 88 Field

The OTHER category for an entity describes a level 88 entry. If a structure line calls for a level 88 entry, the definition is retrieved from the OTHER category. Prefixing of level 88 entries can be suppressed through the OPTIONS statement.

Initial Value Field

The initial value for an element can be stored in the Catalogue. The Generate function retrieves the value from the initial value field in the ATTRIBUTES category for the element and generates a VALUE clause for the Data Division definition.

Justification or Synchronization Indicator Field

The entry for an element can specify that it is justified right or left, or that it is synchronized. In either case, the Generate function generates the appropriate clause for the Data Division definition.

Redefines Field

A structure line in a group entry can specify that the item redefines a previous item in the group structure. Data Catalogue generates the appropriate clauses for the Data Division definition; the redefined definition assumes the hierarchical level of the original item.

Indexed By Field

A structure line in a group entry can specify that the data item is indexed by another data item. Data Catalogue generates an INDEXED BY clause for the Data Division definition using the data name specified in the indexed by field. The item identified by the specified data name is not retrieved from the Catalogue unless it is defined in a separate structure line.

Depending On Field

A structure line in a group entry can specify that the data item depends on another data item. Data Catalogue generates a DEPENDING ON clause for the Data Division definition using the data name specified in the depending on field. The item identified by the specified data name is not retrieved from the Catalogue unless it is defined in a separate structure line.

Occurs From and Occurs To Fields

The occurs from and occurs to fields are specified in a structure line to indicate a repeating data item. An item that occurs a fixed number of times uses only the occurs to field and Data Catalogue generates the OCCURS n TIMES clause for the Data Division definition. An item that occurs a variable number of times uses both fields

and Data Catalogue generates the OCCURS FROM m TO n TIMES clause.

Definition Errors

Sometimes errors exist in the entries stored in the Catalogue. Generation of the listing only for a Generate request provides the means to ensure that the output is correct before the sequential file is created. For example, the PICTURE clause can be checked to determine whether the field length and the picture agree. Care should be exercised in reviewing and changing Catalogue entries.

Generate Request Output

A listing of all requested generations is produced for a Generate request unless the listing has been suppressed by the OPTIONS statement. A sequential file containing the requested generations is also produced unless it has been suppressed by the OPTIONS statement.

PL/I CONSIDERATIONS

When the Generate function is used to create declarative structures for a PL/I program, several considerations should be reviewed. These considerations are discussed in the following paragraphs.

Basis for Generation

The Generate function uses the structure lines of file, record, and group entries to create a PL/I structure. If no structure lines are defined, no generation is possible. Any named structure line must have a matching entry so that the generation process can be carried down the structure for every item.

As a structure line is followed through to the elementary level, the picture, format, length, and other attributes are retrieved from the Catalogue. The relative hierarchical position of each entry within the declarative structure being generated is maintained by the Generate function.

Records are normally assigned level 1; groups and elements that comprise the record are assigned hierarchical levels that descend down the hierarchy in increments of two. Both the starting level and the increment can be changed by the OPTIONS statement.

Identifier Selection

Identifiers are applied to the data entries, each at its appropriate level. Element, group, record, and file entries have the PL/I identifier stored in the NAMES category for the entry. If no identifier is in the entry, the Catalogue name for the entry is used; if the Catalogue name consists of 32 characters, the last character is truncated.

If the PREFIX1 option is selected by the OPTIONS statement, the specified prefix is inserted in front of the identifier retrieved from the PL/I identifier field before completing the generation process. The entire prefix, which can consist of up to nine characters, is placed in the high order positions of the identifier. If the resultant identifier exceeds 31 characters, the low order characters are truncated.

If the PREFIX2 option is selected by the OPTIONS statement, a second set of structures is generated. The identifiers for the second set of structures are generated in the same manner as the first set, except that the PREFIX2 characters are inserted instead of the PREFIX1 characters.

The SUFFIX1 and SUFFIX2 options can be selected by the OPTIONS statement rather than the prefix options. Identifiers are generated in the same manner, except that the suffixes are appended to the end of the identifiers.

Data Attributes

Data attributes are created only for elements. They are derived from the format field and PICTURE clause field. If the PICTURE clause field is defined, it is generated as defined; otherwise, the format field determines the generated picture as follows:

If the format field specifies character format, the generated picture is CHAR(length). The length field value is used for length in the picture.

If the format field specifies numeric format, the generated picture is PIC 9(length), where length is retrieved from the length field.

When the format field for the element specifies binary or hexadecimal format, the PICTURE clause format can specify two integers for precision representation. If the PICTURE clause field is not defined, only single integer precision is defined.

FILLER Entries

Undefined space in a PL/I structure is defined to Data Catalogue as FILLER. The picture generated for filler items is always defined as CHAR(length), where length is determined by the length field in the entry.

Other Entity Fields

A group or record entity can include the occurs from and occurs to fields in a structure line. If these fields are specified, they are generated as array dimensions for the specified group or element. Storage attributes defined are generated at the appropriate level.

Compiler Editing

No compiler level editing is performed by Data Catalogue. Only the default values specified are annotated. Any inconsistencies between pictures and lengths are not noted.

TOTAL DBDL CONSIDERATIONS

DBDL can be generated from Data Catalogue entries. The generation process begins by retrieving a specified TOTAL data base entry from the Catalogue. The STRUCTURE category in the data base entry is then examined.

The data base entry provides a name for the data base and the names of each master and variable dataset found in the data base. If no structure lines exist, no DBDL can be generated.

The master or variable datasets are retrieved in turn. Dataset attributes are stored for DBDL entries. Once the dataset attributes have been formatted, the structure lines of the dataset are accessed. These lines point to component records in the dataset.

The component records are accessed and record structure lines are examined. Base data portions of a record are retrieved by group and element, and the length of each data element and its TOTAL name are retrieved from the appropriate entry.

If an element is a key item or a linkage path item, it is so coded in the record structure line. A Key indicator or Linkage Path entry is generated. If a record contains

record codes, these are defined by a record code indicator in the structure lines. The component portions of the variable portion of a record code item are retrieved from their respective group and element entries, and the length and structure of the element and data item are formatted for DBDL.

Basically, the TOTAL data base entry is exploded from data base to data item to retrieve the necessary attributes for DBDL. In addition, the dataset furnishes DBDL environmental requirements including the logical record length, total logical records, and so forth. The structure must be defined correctly to Data Catalogue for DBDL to be generated correctly.

The process of creating Data Catalogue entries can be simplified by the use of the Convert function. This function scans an existing COBOL module and derives a set of transactions that can be used to create Data Catalogue entries. It can also read an existing TOTAL DBDL and create input transactions for Data Catalogue. The Convert function is a batch processing operation.

CONVERT APPLICATIONS

Various goals can be accomplished through the Convert function. These goals are as follows:

- Create entry definitions in transaction input format.
- Obtain and store module usage of data in Data Catalogue.
- Obtain analysis of a set of modules in terms of common data entities.
- Obtain a checklist of data entities to be described in a specified application.

CREATE ENTRIES

The Convert function can be used to create a series of Data Catalogue transactions to be input to the Update function for creating new entries. This is usually performed by loading the new entries onto a working Catalogue. The Query function and the Report function can then be used to analyze the entries before making them a part of the permanent Catalogue.

An entry created through the Convert function is skeletal because the only available data is that which is described in the COBOL module or the data base definition. This information can vary from the minimum of name, length, format, and picture up to a record with its name, length, structural components, redefined items, variable occurrences, indexed by items, and depending on items.

The recommended method for creating entries is to store them in a working Catalogue first. The entries can then be updated by adding description lines, usage data, values, origin, and other attributes. The updated entries can then be transferred to the permanent Catalogue by a utility available to the data administrator.

DATE USAGE IN MODULES

Module usage of data structures can be obtained by the Convert function from COBOL FD entries. This can be very useful in creating linkage entries that can be used in determining which other modules are affected by a change to data used by that module.

The level of usage to be stored can be specified to the Convert function. Initially, only the file level usage could be stored; for example, a relational line for the module could indicate that it uses FILE-A. The record level usage could be added, and so on down the hierarchy.

The type of usage obtained by the Convert function is defined as being present in the specified module. This can be useful when processing modules that define only the data with which they are concerned.

ANALYSIS OF MODULES

The Convert function can be used to analyze a set of modules that process common data. The COBOL FD entries are used for the analysis. The consolidation report shows elements that are defined only once. By implication, this identifies those modules that are not using common definitions. The use of the Convert function as a general purpose analyzer of data structures might be beneficial in many data administration projects.

The end product of the analysis process includes an alphabetized listing of all the elements, groups, records, and files processed. The names in this list can be scanned to determine which modules are the worst offenders of name usage. These modules can then be noted for recompilation using preferred names at the next time the modules come up for maintenance.

CHECKLISTING DATA

The Convert function can also be used to check on the definition of a data structure that has been created independently. For example, a file structure entered through a terminal could be checked to determine that the described structure is complete.

The DBDLs or modules that process a particular data structure are passed to the Convert function. The resulting analysis report can then be reviewed for completeness of the data structure. Structure lines for the data structures created by the Convert function must parallel the independently-entered data structure. If the conversion analysis report shows additional entities, an element or group could have been omitted in the independently-entered data structure. The report can be traced down by element and group until the structure is complete.

CONSIDERATIONS FOR CONVERSION

The Convert function provides the means to obtain data definitions and structures through an analysis of existing modules or DBDLs. The data definitions within the modules or DBDLs can be scanned; specific data definitions can be selected, analyzed, and formatted into transactions that can be input to the Update function to create new Catalogue entries.

This very powerful ability must be used with care. Many factors must be considered before using the Convert function. Great value can be gained from the use of the Convert function once all the factors have been weighed and the DBDLs or modules have been determined to be appropriate sources for definitions. The main factors to be considered are as follows:

Thoroughness of definitions in DBDLs or modules

Commonality of names in modules

Number of modules to be scanned

Coding techniques used in modules

THOROUGHNESS OF DEFINITIONS

The first factor to be considered in automating the conversion of existing DBDLs or modules is the quality of the source. If a module describes a file or record only to the extent it uses data, the resulting definitions are only a subset of the complete file or record definition. It might be necessary to scan many different modules to obtain a complete description of all elements within the file or record. If a module describes a file or record in its entirety, however, only one module scan is necessary for a complete description.

Thoroughness of definition is not always an asset. If elements or records are defined but are completely unused or are obsolete, it is not useful to store these descriptions in the Catalogue. The resource should be reviewed before the scanning process is started to ensure that the entries created from the Convert function transactions are all current and usable definitions of data.

COMMONALITY OF NAMES

A file or record can be described in several modules. If common names are used in all modules, the Convert function consolidates the descriptions into one entry and records the usage of the data in each of the modules processed. Many times, the names are different from module to module. In this case, the Convert function cannot perform automatic consolidation of the descriptions and creates separate entries for each different name unless the renaming facility is used.

The Convert function has the ability to rename data structure definitions. This makes it possible to assign names to data entries. If it is necessary to rename almost all of the data entered into the Convert function, the amount of preparatory work is extensive.

NUMBER OF MODULES TO BE SCANNED

The number of modules that must be scanned to obtain a definition should be kept as low as possible. When it is necessary to process dozens of modules to produce a definition and obtain some usage data, processing costs and analytical time required to complete the process are increased. Target modules should be carefully selected to obtain maximum benefit from a minimum quantity of data.

The Convert function creates Data Catalogue entries for existing COBOL programs and TOTAL DBDLs. The program or DBDL is scanned for selected structures and Update function input transactions are generated for the new entries. The Convert function output must be carefully reviewed, particularly when converting more than one module at a time.

When a selected record appears in several modules, the Convert function assumes that multiple record layouts can exist. The function, therefore, includes all the layouts in the STRUCTURE category for the record. Figure 8-1 illustrates two different layouts for the record PRINT-LINE and the resulting structure lines created by

the Convert function. In this example, both record layouts are included in their entirety; fields with the same data name appear twice. It is the user's responsibility to format the STRUCTURE category in the manner the record is desired to be represented in the Catalogue.

<u>Layout in REC-A</u>	
01	PRINT-LINE.
03	FIELD-1 PIC X(3).
03	FIELD-2 PIC X(8).
03	FIELD-3 PIC X(2).
03	FIELD-4 PIC X(2).
03	FIELD-5 PIC X(5).
<u>Layout in REC-B</u>	
01	PRINT-LINE.
03	FIELD-1 PIC X(3).
03	GROUP-1.
05	G-FIELD-1 PIC X(3).
05	G-FIELD-2 PIC X(5).
03	FILLER PIC X(4).
03	FIELD-5. PIC X(3).
<u>Structure Category for PRINT-LINE</u>	
0010	CATNAME=FIELD-1
0015	CATNAME=FIELD-2
0020	CATNAME=FIELD-3
0025	CATNAME=FIELD-4
0030	CATNAME=FIELD-5
0035	CATNAME=FIELD-1
0040	CATNAME=GROUP-1
0045	CATNAME=FILLER
0050	CATNAME=FIELD-5

Figure 8-1. Conversion with Multiple Record Layouts

CODING TECHNIQUES

Coding techniques used in the modules selected for the Convert function should also be reviewed. The Convert function assumes that modules describe files and records in the conventional manner. If all files, records, and hierarchical data structures are conventionally defined within the module, for example, the Convert function can process the target module successfully. When records are passed to a module and then redefined in the module with a set of names purely internal to the module, nothing can be gained by processing that module.

CONVERT REQUEST STRUCTURE

A Convert request consists of a series of statements that identify DBDLs or modules to be scanned and target data to be converted. The type of output to be produced by the Convert request can be controlled by additional statements.

Statements applicable to a Convert request are as follows:

\$CONVERT

Identifies the request as a Convert request and specifies the type of conversion.

MODULE

Identifies a module to be scanned; not applicable for a TOTAL DBDL.

SELECT

Identifies a module data structure or a DBDL to be converted.

RENAME

Assigns a Catalogue name to an entry being created; also used for aliases and versions.

CREATE

Controls the creation of Data Catalogue transactions and defines the level of entries to be created.

A Convert request is initiated by a \$CONVERT statement. When a module is being converted, the MODULE statement is required. At least one SELECT statement must be included for each DBDL or module to be converted. The RENAME and CREATE statements are optional.

Figure 8-2 illustrates a typical deck setup for a COBOL module Convert request. Figure 8-3 illustrates a typical deck setup for a TOTAL DBDL Convert request.

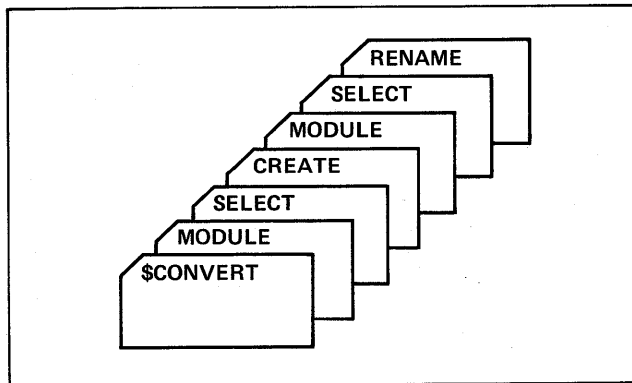


Figure 8-2. Convert Request Deck Setup, COBOL Conversion

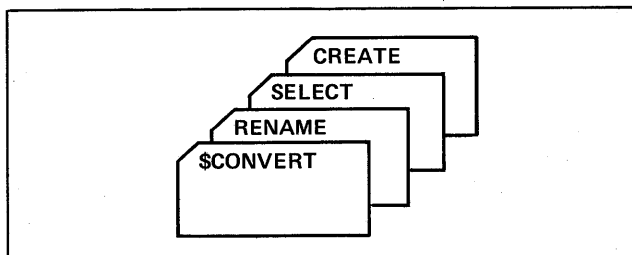


Figure 8-3. Convert Request Deck Setup, TOTAL DBDL Conversion

\$CONVERT STATEMENT

A Convert request is initiated by the \$CONVERT statement. The type of conversion requested is also specified in this statement. The format of the \$CONVERT statement is shown in figure 8-4.



Figure 8-4. \$CONVERT Statement Format

Phrases in the \$CONVERT statement are defined as follows:

COBOL

Specifies that a COBOL module is to be converted.

DBDL

Specifies that a TOTAL DBDL is to be converted.

MODULE STATEMENT

The COBOL module to be scanned for conversion is specified by the MODULE statement. Multiple statements are included in the Convert request when multiple modules are to be scanned. The format of the MODULE statement is shown in figure 8-5.

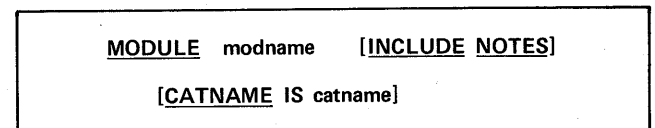


Figure 8-5. MODULE Statement Format

Phrases in the MODULE statement are defined as follows:

modname

Specifies the name of the module to be scanned for conversion. This is the PROGRAM-ID name in the COBOL program.

INCLUDE NOTES

Requests the formatting of comment lines into transactions for the Update function.

CATNAME IS catname

Specifies the Catalogue name for the module entry to be created.

The following rules are applicable to the MODULE statement:

At least one MODULE statement must be included for a COBOL Convert request.

The module to be scanned must be identified by the modname phrase.

Comment or note lines requested through the INCLUDE NOTES phrase are placed in a group of lines immediately following the structure lines that describe the record being converted.

When only groups and elements are being converted, no comment lines are converted because these are only stored at the record level.

Any catname specified must conform to the rules for forming Catalogue names.

SELECT STATEMENT

The SELECT statement for a Convert request identifies a DBDL or data structure to be converted. At least one SELECT statement must be included for each DBDL or module to be converted. Multiple SELECT statements can be specified for one MODULE statement. The format of the SELECT statement is shown in figure 8-6.

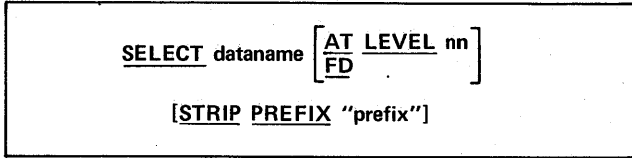


Figure 8-6. SELECT Statement Format, Convert Function

Phrases in the Convert request SELECT statement are defined as follows:

dataname

Specifies the name of the data structure to be converted. For COBOL, this is the data name; for TOTAL, this is the DBDL name.

AT LEVEL nn

Specifies the level number for the specified COBOL data name; default is level 01; not applicable to a TOTAL DBDL.

FD

Specifies a COBOL file level (FD) is to be converted; not applicable to a TOTAL DBDL.

STRIP PREFIX "prefix"

Requests deletion of specified prefix characters from the COBOL data name; not applicable to a TOTAL DBDL.

The following rules are applicable to the SELECT statement:

One SELECT statement must be included for each DBDL or module to be converted. Any number of SELECT statements can follow a MODULE statement.

A prefix to be stripped must specify all characters to be stripped from the name including the hyphen if it is part of the prefix.

RENAME STATEMENT

The optional RENAME statement is used to assign a Catalogue name to a data entry being created by the Convert function. It can also be used to indicate that the entry is an alias or version of an existing entry. The format of the RENAME statement is shown in figure 8-7.

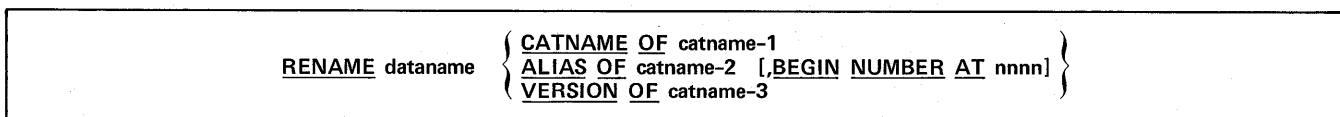


Figure 8-7. RENAME Statement Format

Phrases in the RENAME statement are defined as follows:

dataname

Specifies a data name in the DBDL or module to be renamed or declared an alias or version.

CATNAME OF catname-1

Specifies the Catalogue name to be assigned to a DBDL or module data name. If this phrase is omitted, the DBDL name or COBOL data name is used as the Catalogue name.

ALIAS OF catname-2

Indicates that the specified entry is an alias of an existing entry, which is indicated by catname-2. The BEGIN NUMBER AT phrase is included for an element that is to be entered in the ALIAS category for the existing entry at the specified alias line number.

VERSION OF catname-3

Indicates that the specified entry is to be set up as a version of an existing entry, which is indicated by catname-3.

The following rules are applicable to the RENAME statement:

A Convert request for a TOTAL DBDL requires the RENAME statement to immediately follow the \$CONVERT statement. The RENAME statement is optional; up to 100 statements can be specified, thus forming a table.

A Convert request for a COBOL module requires the RENAME statement to immediately follow the SELECT statement to which it refers. The RENAME statement is optional; up to 100 statements can follow a SELECT statement.

If the RENAME statement specifies the CATNAME OF phrase, each occurrence of dataname found in the selected data structure or DBDL is assigned the specified Catalogue name.

If a RENAME statement is not specified, the module data name or DBDL name becomes the Catalogue name.

If the ALIAS OF or VERSION OF phrase is specified, the new entry is set up with the dataname as an alias or version of catname-2 or catname-3.

The BEGIN NUMBER AT phrase specifies that dataname is set up in the ALIAS category of catname-2.

CREATE STATEMENT

The CREATE statement can be used to specify which entity types are to be created by the Convert function. If the CREATE statement is not included in the Convert request, all entity types are processed. Each SELECT statement can have an associated CREATE statement, or several SELECT statements can use the same CREATE statement. The options selected by a CREATE statement are in effect until another CREATE statement is encountered. The format of the CREATE statement is shown in figure 8-8.

```
CREATE [ELEMENTS] [,GROUP] [,RECORDS]
      [,FILE] [,PROGRAM] [,DATASETS]
      [,DATABASE]
```

Figure 8-8. CREATE Statement Format

Phrases in the CREATE statement are defined as follows:

ELEMENTS

Creates transactions for element entities only.

GROUPS

Creates transactions for group entities only.

RECORDS

Creates transactions for record entities only.

FILE

Creates transactions for file entities only; not applicable to TOTAL.

PROGRAM

Creates transactions for program (module) entities only; not applicable to TOTAL.

DATASETS

Creates transactions for dataset entities only; not applicable to COBOL.

DATABASE

Creates transactions for data base entities only; not applicable to COBOL.

The following rules are applicable to the CREATE statement:

If no CREATE statement is encountered, the entities generated are as follows:

For COBOL, element, group, record, file, and program entities are created.

For TOTAL, element, group, record, dataset, and data base entities are created.

Only one CREATE statement can be entered for a SELECT statement.

Phrases identifying entity types can be specified in any order.

The Data Catalogue Utility function is provided to assist the data administrator in performing the tasks necessary to support the Data Catalogue files. This section presents an overview of the Utility function to provide an understanding of the supporting facilities. A detailed description of this function is contained in the data administrator's reference manual.

The following major facilities of the Utility function are briefly described in this section:

- System initialization
- System specialization
- Common utilities

SYSTEM INITIALIZATION

The first step to be performed in using Data Catalogue is to initialize the system. This involves identifying the installation and initiating system files and certain system features. The data administrator is usually responsible for this procedure.

INSTALLATION IDENTIFICATION

The installation must be identified to Data Catalogue by name and address. This is accomplished by two transactions. The name and address are used as heading material for all standard Data Catalogue reports.

The system environment must also be specified at initialization time. The number of home blocks to be preallocated for the direct access files to hold system files must be specified.

These two operations must be performed at system initialization. Optional features can also be selected.

OPTIONAL FEATURES

Certain optional features can be selected at system initialization. These features include the following:

- Implementation standards
- Report options

Each of these features is implemented by one or more transactions. Table 9-1 lists the options and their default values. Whether or not any option is selected, the initialization process must still be performed.

The implementation standard that can be specified at system initialization is the requirement of certain fields for Catalogue entries. When a required field is deleted during maintenance of Catalogue entries, an informative message is produced.

Report options that are selected at system initialization are as follows:

Page overflow

The standard setting of 59 lines to a page can be changed to the installation setting.

Footnote message

A message to be printed at the bottom of each standard Data Catalogue report page can be specified. The specified message is printed in the bottom right corner of the report page.

TABLE 9-1. INITIALIZATION OPTIONS

Feature	Default
Implementation standards	No user-defined standards are defined for any fields.
Report options: Lines per page	Page overflow occurs at line 59.
Footnote message	No footnote message is printed at the bottom of each standard report page.

OPTIONS MAINTENANCE

After the Data Catalogue system has been initialized, system options can be changed through the system maintenance facility of the Update function. This includes the following capabilities:

- Display existing system options
- Change existing options by adding, changing, or deleting options

OPTIONS DISPLAY

Options in effect at any given time can be displayed. The display presents a response showing the name, address, implementation standards, and report options in effect at the time the request is initiated.

OPTION CHANGES

The specification for any option can be changed at any time. Optional features can be added, changed, or deleted through a set of transactions.

SPECIAL FACILITIES

Several special facilities can be performed by the data administrator to maintain the Catalogue. The Utility function can be invoked to perform the following operations:

Transfer definitions from a working Catalogue to a permanent Catalogue

Backup and restore system files

Display system statistics

Rename entries

Re-number lines in entry categories

DEFINITION TRANSFER

Two separate Catalogues of definitions can be developed. One is used for developing definitions and is usually referred to as the test Catalogue. The other contains the finished definitions and is usually referred to as the production Catalogue. Other Catalogues can be developed to meet divisional needs or to reflect important installation considerations.

Entries can be moved from one Catalogue to another. At this time, an entry can be renamed and it can be retained in or deleted from the original Catalogue as the transfer occurs. Transfers can be specified for an element or a higher-level entry, which transfers the higher-level entry and its lower-level entries.

BACKUP AND RESTORE

Backup and restore facilities are provided to ensure that system files are not lost. The backup facility provides the means to save system files and accumulated control statistics. A listing is also produced by the backup facility for documentation purposes.

The restore facility uses the file produced by the backup facility to recreate the system files. As the restore facility rebuilds the system files, it also maintains a count of records created by entity type and a count of physical

records. These totals can be used to reconcile the totals from the backup facility.

SYSTEM STATISTICS

Data Catalogue maintains records of current statistics. The data administrator is provided with pertinent information through these statistics. Information provided is as follows:

Current revision number for the Catalogue

Date of last update

Date of last restore and number of records affected

Date of last backup and number of records affected

Number of updates in the last run

Number of inserts in the last run

Number of deletes in the last run

LINE RENUMBERING

Lines in an entry can be renumbered; this can apply to all lines or to lines in a specific category. Renumbering occurs most commonly in the DESCRIPTION, VALUES, and other categories utilizing multiple lines.

RENAMING ENTRIES

Entries can be assigned new Catalogue names through the Utility function. The Catalogue name can be changed for the entry and in every entry that references the Catalogue name, or it can be changed only for the entry. All references to the previous Catalogue name can also be deleted.

Data Catalogue is implemented through six functions. The modules containing the executable programs for these functions are stored on various permanent files that must be made available to the functions. The Catalogue master files are also permanent files that must be made available to the functions. In addition, some functions require work files and sort operations.

This section discusses the file requirements and operational considerations for using Data Catalogue. Following a general discussion, specific requirements and considerations are presented for each Data Catalogue function.

COMMON CONSIDERATIONS

All Data Catalogue functions require certain permanent files to perform the functions. Pseudo-switch settings are also used by Data Catalogue.

EXECUTABLE PROGRAMS

At least one module containing executable programs is required to perform a Data Catalogue function. These modules are stored on the system as permanent files. The permanent file names of the modules for each function are as follows:

Utility	DCUTL
Update	DCUPD
Query	DCRET
Report	DCSEL, DCRPT, and DCIDX
Generate	DCRET
Convert	DCCONVT and DCCONGN

PSEUDO-SWITCH SETTINGS

Four pseudo-switches are used by Data Catalogue. The first pseudo-switch is set by the user to indicate the processing mode. The other switches are set by Data Catalogue to indicate the function termination condition. The same switches are used by all functions. The pseudo-switches are defined as follows:

SW1	Set to ON for interactive mode; applicable only for the Update, Query, and Generate functions. If it is not set to ON before execution of DCUPD or DCRET, the function operates in batch mode.
SW2	Set to ON for a normal termination of the function.
SW3	Set to ON for a normal termination; however, an error occurred during processing. The Catalogue master files are not impaired, but the output should

be carefully reviewed.

SW4

Set to ON for an abnormal termination; a fatal error occurred during processing. The Catalogue master files were closed at the point the error was detected. The output should be reviewed to determine the nature of the error. It might be necessary to restore the files.

COMMON FILES

Each function uses the Catalogue master files. In addition, some functions require work files. For card image input and printed output, the default files are the system files INPUT and OUTPUT, respectively.

The Catalogue master files are three permanent files that contain all the information related to the Catalogue entries. These three files are referred to collectively as the Catalogue. Each new Catalogue that is initialized through the Utility function has its own copy of the master files. The three files are the data file, the relational file, and the control file. Table 10-1 lists the file characteristics of the master files.

TABLE 10-1. CATALOGUE FILE CHARACTERISTICS

Characteristic	Data File	Relational File	Control File
Logical file name (LFN)	MAST1	MAST2	MAST3
File organization (FO)	Direct access	Direct access	Word addressable
Maximum block length (MBL)	2550	1270	3840
Maximum record length (MRL)	1265	620	3830
Minimum record length (MNR)	1265	620	3830
Record type (RT)	F	F	F
Key length (KL)	36	36	--
Relative key word (RKW)	0	0	--
Relative key position (RKP)	0	0	--
Home blocks (HMB)	User-defined†	User-defined†	--
†By data administrator.			

FUNCTIONAL CONSIDERATIONS

The file requirements and operational considerations are different for each Data Catalogue function. Input/output operations for all functions are performed by CYBER Record Manager (CRM). Error messages and file statistics are output by CRM to an error file when FILE control statements for the Catalogue files specify EFC=3. The CRMEP control statement is then used to print the messages and statistics.

UTILITY FUNCTION

System initialization is performed through the Utility function; this is the first Data Catalogue function that is run. The Utility function is also used after initialization of the system to perform other support type operations.

The Utility function is a data administrator function; therefore, the operating requirements are detailed in the data administrator's reference manual. A summary of the capabilities of the function is contained in section 9.

UPDATE FUNCTION

Maintenance of the Catalogue is performed through the Update function. Entries or portions of entries can be added, changed, or deleted by this function. The Update function is described in detail in section 4.

Required Files

The Update function is implemented by execution of the module on file DCUPD. All three files in the Catalogue (MAST1, MAST2, and MAST3) are required. The Update function also requires card image transactions from the system file INPUT, and it outputs reports to the system

file OUTPUT. A local work file, TEMP1, is transparent to the user. The Update Audit Report is not produced when the Update function is used interactively. Figure 10-1 shows the file usage flow for the Update function.

Update transactions are read from system file INPUT. The first two transactions must be the \$UPDATE statement and the OPTIONS statement. Detail transactions then follow the OPTIONS statement.

The files MAST1, MAST2, and MAST3 are created as part of the initialization process; therefore, they only need be attached before execution of the DCUPD module. Figure 10-2 shows the control statements required for execution of the Update function.

Operational Considerations

The Update function requires a minimum field length of 64000g words. The programs that constitute the DCUPD module are overlaid during processing.

The setting of pseudo-switch SW1 determines whether the Update function is performed in batch or interactive mode of operation. The switch must be set before the DCUPD module is called into execution. For interactive mode, SW1 must be set to ON; for batch mode it must be set to OFF. Under the NOS/BE operating system, the INPUT and OUTPUT files must be connected before interactive execution of DCUPD. A %EOR is used to terminate INPUT directives.

Data need not be sequenced by key. No advantage is gained by such sequencing from an operational view.

Data must be sequenced in logical order. An entry must be added before it can be changed. If an entry is to be deleted and then replaced with new information for the

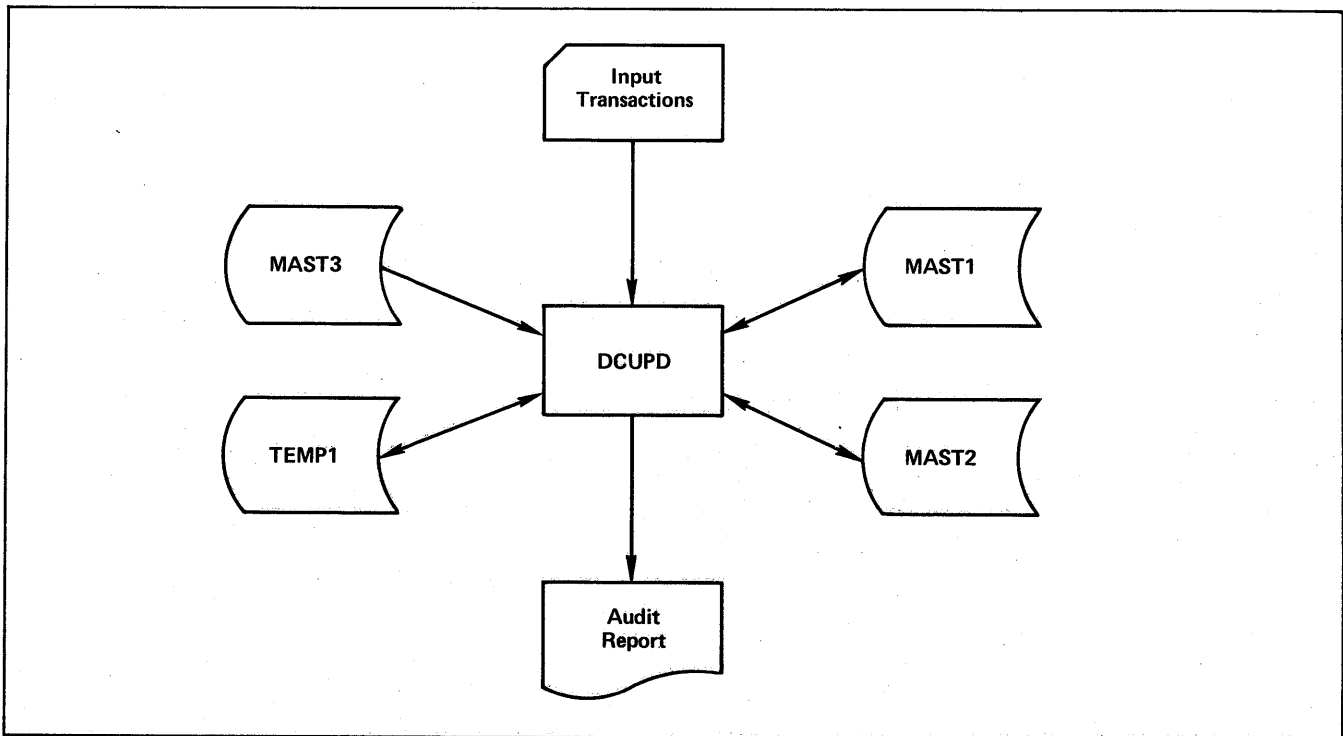


Figure 10-1. Update Function File Usage

NOS Operating System

Job statement
USER control statement
CHARGE control statement
ATTACH,DCUPD/UN=LIBRARY.
ATTACH,MAST1,MAST2,MAST3/M=W.
FILE,MAST1,FO=DA,EFC=3.
FILE,MAST2,FO=DA,EFC=3.
FILE,MAST3,FO=WA,EFC=3.
DCUPD.
CRMEP,LO.

NOS/BE Operating System

Job statement
ATTACH,DCUPD,ID=LIBRARY.
ATTACH,MAST1,ID=name
ATTACH,MAST2,ID=name
ATTACH,MAST3,ID=name
FILE,MAST1,FO=DA,EFC=3.
FILE,MAST2,FO=DA,EFC=3.
FILE,MAST3,FO=WA,EFC=3.
DCUPD.
CRMEP,LO.

Figure 10-2. Update Function Control Statements

same Catalogue name, the entry must first be deleted and then be added to the Catalogue.

When creating a new entry, throughput is greatly enhanced if data is entered in the same category sequence listed in appendix D.

Run time options are selected by the OPTIONS statement. This statement is required to specify whether revision numbers are to be checked. Other options can also be selected. Selected options are in effect only for the specified Update request.

Output

The three master files of the Catalogue are updated by the Update function. For the first few months of operation, it is recommended that the files be backed up and restored on a frequent basis for protection. The Utility function or operating system copy utilities can be used for backing up and restoring files.

In batch mode only, printed output from the Update utility includes the Update Audit Report. This is a transaction listing with error messages. The last page of the audit report is a series of statistics providing entry and transaction statistics. The final output, which indicates the termination of the Update function is as follows:

```
*** END AUDIT REPORT ***
```

QUERY FUNCTION

The Catalogue can be interrogated through the Query function. The queries can be generalized or specific; responses can be summarized or detailed. The Query function is described in detail in section 5.

Required Files

The Query function is implemented by execution of the programs in the DCRET module. The entire Catalogue (files MAST1, MAST2, and MAST3) is required for this function. Card image input transactions are also required. The Query function outputs the response report to the system file OUTPUT. Figure 10-3 shows the file usage flow for the Query function. The control statements required to execute the Query function are shown in figure 10-4.

Operational Considerations

The minimum field length requirement for the Query function is 70000g words. The programs that constitute the function are overlaid during processing.

Pseudo-switch SW1 must be set to ON before the DCRET module is called into execution for interactive mode of operation. For batch mode of operation, switch SW1 must be set to OFF before the DCRET module is called into execution. Under the NOS/BE operating system, the INPUT and OUTPUT files must be connected before interactive execution of DCRET. A %EOR is used to terminate INPUT directives.

The Query and Generate functions both require execution of the DCRET module; therefore, these functions can be combined in one call for execution of DCRET. The functions can be requested in any order. No sequencing by key fields is required.

A Query request searches the Catalogue for qualified entries. Requests that do not specify search criteria take longer than those that limit the search to entries within a hierarchy. For example, a query for a given character string requires all entries of the specified type to be searched unless the Query request limits the search to entries that meet a specified criteria. If the request for the character string search specifies element entity types, all element entries are searched; however, if the search is limited to elements within a given hierarchy, the search time is reduced. The fastest queries are those addressed to specific entries.

The Query function has no run time options governing operational considerations.

Output

The Query function outputs responses to the Query requests. These responses are either detailed or summarized based on the statement specifications in the Query request. A COUNT statement produces a one-line response regardless of the number of qualifying entries. A LIST statement produces a one-line response for each qualifying entry. A SHOW statement can produce a page or more for each qualifying entry.

The final output for the Query function is as follows:

```
*** TOTAL QUERY COMMANDS ATTEMPTED = n  
*** TOTAL COMMANDS ACCEPTED = n  
*** TOTAL COMMANDS REJECTED = n
```

After all Query and Generate requests have been performed, the following message is output:

```
*** END DATA CATALOGUE RETRIEVAL REPORTS ***
```

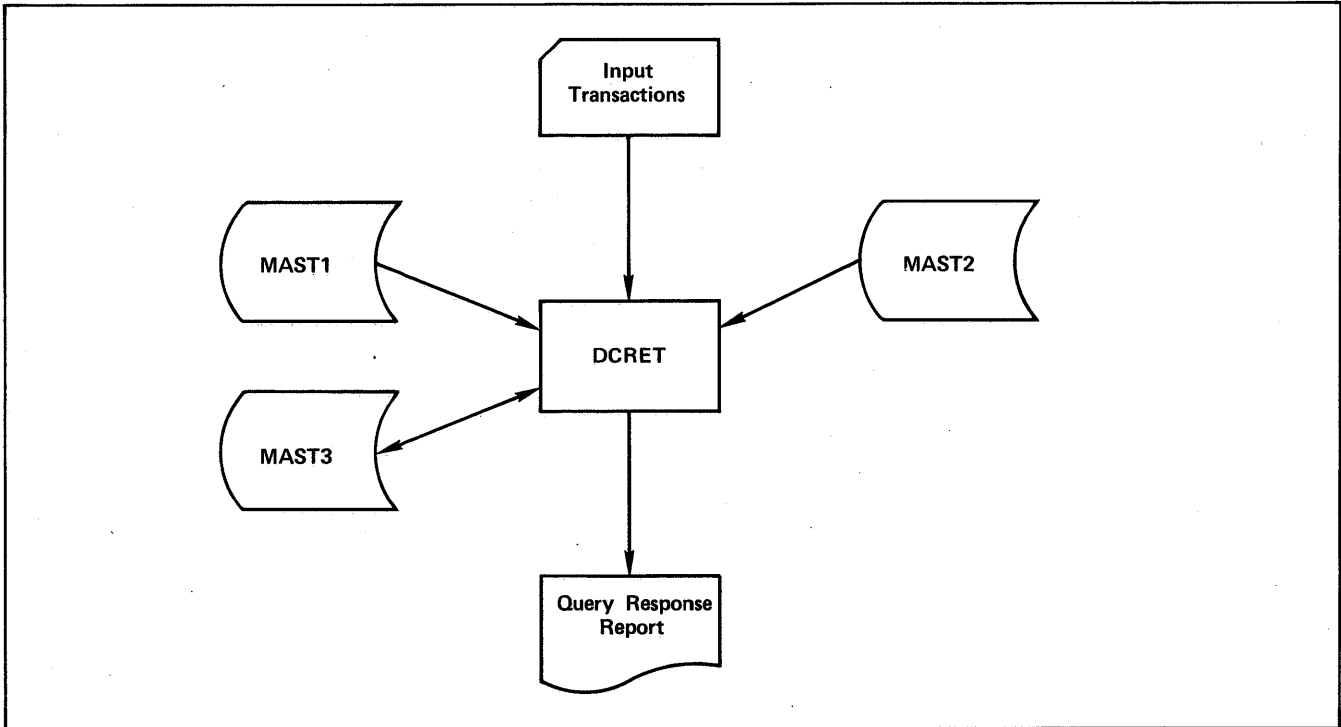


Figure 10-3. Query Function File Usage

NOS Operating System

Job statement
 USER control statement
 CHARGE control statement
 ATTACH,DCRET/UN=LIBRARY.
 ATTACH,MAST1,MAST2,MAST3/NA.
 FILE,MAST1,FO=DA,EFC=3.
 FILE,MAST2,FO=DA,EFC=3.
 FILE,MAST3,FO=WA,EFC=3.
 DCRET.
 CRMEP,LO.

NOS/BE Operating System

Job statement
 ATTACH,DCRET,ID=LIBRARY.
 ATTACH,MAST1,ID=name.
 ATTACH,MAST2,ID=name.
 ATTACH,MAST3,ID=name.
 FILE,MAST1,FO=DA,EFC=3.
 FILE,MAST2,FO=DA,EFC=3.
 FILE,MAST3,FO=WA,EFC=3.
 DCRET.
 CRMEP,LO.

Figure 10-4. Query Function Control Statements

REPORT FUNCTION

Various reports can be produced through the Report function. These reports present various types of detailed information in either 8-1/2 by 11 or computer-width format. The Report function is described in detail in section 6.

Required Files

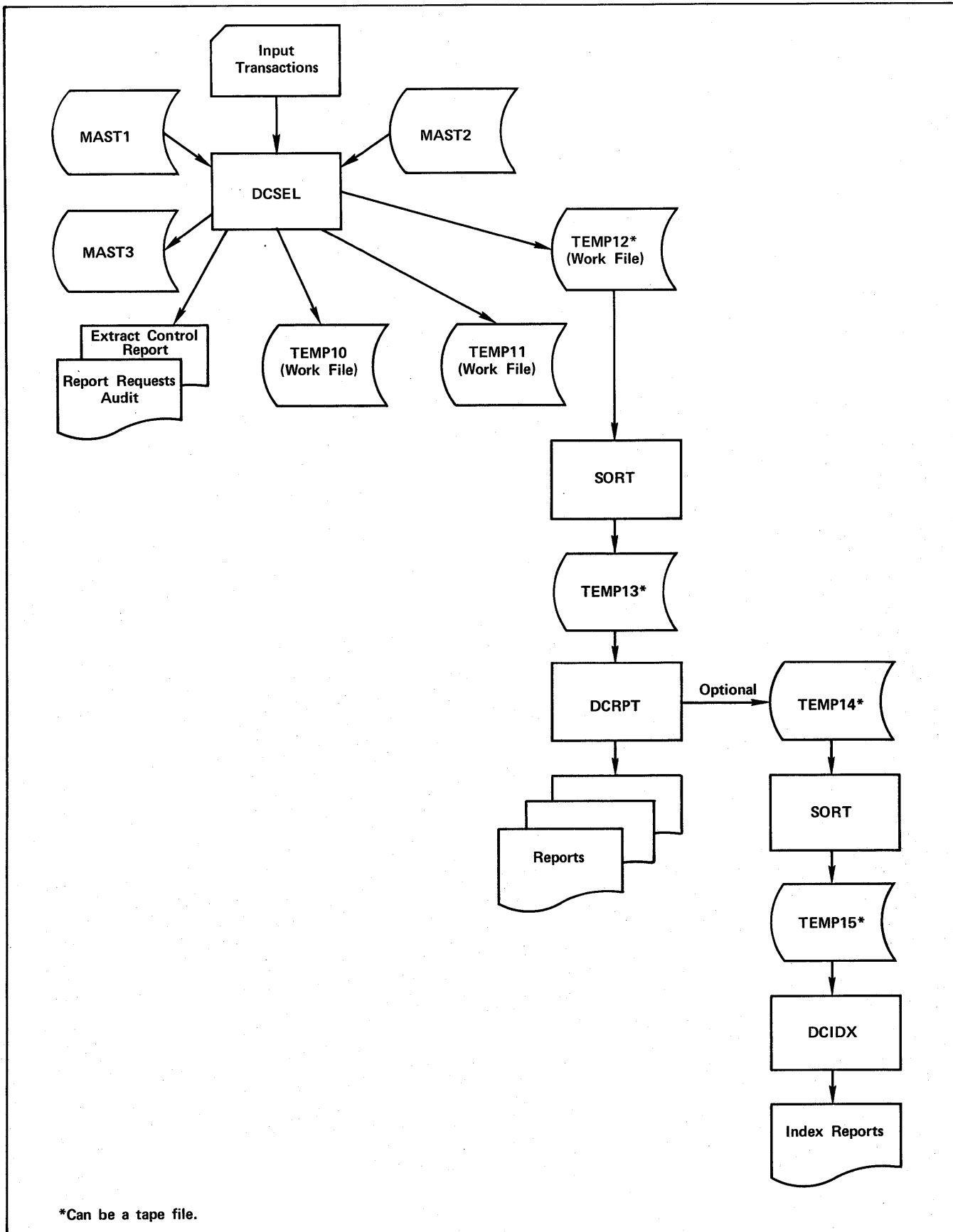
The DCSEL module must be executed to extract the report data. A sort operation and execution of the DCRPT module are required to produce the report. If an index for the report is requested, another sort operation and execution of the DCIDX module are also required.

All three files in the Catalogue (MAST1, MAST2, and MAST3) are required for the Report function. Other required files are the system file INPUT for input transactions, the system file OUTPUT for printed reports, and six temporary files (TEMP10 through TEMP15). Figure 10-5 shows the file usage flow for the Report function. The control statements required to execute the Report function are shown in figure 10-6.

Three temporary files (TEMP10, TEMP11, and TEMP12) are created by the DCSEL module. All valid report requests are written on a local work file, TEMP10. TEMP11 is also a local work file used by DCSEL.

Data selected from the Catalogue is written on TEMP12, which the user can direct to a system internal formatted tape or to disk. TEMP12 can be made a permanent file to be used in a later run, or it can be used as a local work file. TEMP12 is a sequential file with C type blocks and W type records. The records vary in size from 100 to 294 characters.

In addition to printing the Report requests, the DCSEL module prints the Extract Control Report. This report provides totals of the records selected by each request and each SELECT statement within a request. The report can be reviewed before performing the next processing step, the sort operation. In this case, TEMP12 must be tape resident or made a permanent file during the DCSEL module processing step.



*Can be a tape file.

Figure 10-5. Report Function File Usage

NOS Operating System

Job statement

```
USER control statement
CHARGE control statement
ATTACH,MAST1,MAST2,MAST3/NA.
FILE,MAST1,FO=DA,EFC=3.
FILE,MAST2,FO=DA,EFC=3.
FILE,MAST3,FO=WA,EFC=3.
ATTACH,DCRPT,DCSEL,DCIDX/UN=LIBRARY,NA.
FILE,TEMP12,BT=C,RT=W,MRL=294,MNR=100,CM=NO.
FILE,TEMP13,BT=C,RT=W,MRL=294,MNR=100,CM=NO.
DCSEL.
REWIND,TEMP12.
SORTMRG(0=VMSG/R)
FILE,TEMP14,BT=C,RT=W,MRL=135,MNR=66,CM=NO.
FILE,TEMP15,BT=C,RT=W,MRL=135,MNR=66,CM=NO.
DCRPT.
REWIND,TEMP14.
SORTMRG(0=VMSG/R)
DCIDX.
CRMEP,LO.
```

NOS/BE Operating System

Job statement

```
ATTACH,MAST1,ID=name.
ATTACH,MAST2,ID=name.
ATTACH,MAST3,ID=name.
FILE,MAST1,FO=DA,EFC=3.
FILE,MAST2,FO=DA,EFC=3.
FILE,MAST3,FO=WA,EFC=3.
ATTACH,DCRPT,ID=LIBRARY.
ATTACH,DCSEL,ID=LIBRARY.
ATTACH,DCIDX,ID=LIBRARY.
FILE,TEMP12,BT=C,RT=W,MRL=294,MNR=100,CM=NO.
FILE,TEMP13,BT=C,RT=W,MRL=294,MNR=100,CM=NO.
DCSEL.
REWIND,TEMP12.
SORTMRG(0=VMSG/R)
FILE,TEMP14,BT=C,RT=W,MRL=135,MNR=66,CM=NO.
FILE,TEMP15,BT=C,RT=W,MRL=135,MNR=66,CM=NO.
DCRPT.
REWIND,TEMP14.
SORTMRT(0=VMSG/R)
DCIDX.
CRMEP,LO.
```

Figure 10-6. Report Function Control Statements

TEMP12 is sorted in the next processing step. The sort operation output is stored on TEMP13. TEMP13 must be defined as a sequential file with C type blocks and W type records. The records contain from 100 to 294 characters. The sort key begins in character position 3, is 88 characters in length, and is in display code. The sort is an ascending sort in the COBOL6 sequence.

The next processing step is execution of the DCRPT module. In addition to writing the requested reports, DCRPT produces two brief reports for each request. The Request Summary Report is printed before each requested report. It reviews the options and parameters defined by the request. It also contains the same totals printed in the Extract Control Report produced by the DCSEL module. The second report produced by DCRPT is the Request Control Report, which provides various record counts; this report is printed following the requested report.

TEMP13, which contains the sort output, is input to the DCRPT module. If index reports have been requested, DCRPT creates a temporary file, TEMP14, which the user can direct to a system internal formatted tape or to disk. TEMP14 can be made a permanent file to be used in a later run, or it can be used as a local work file. TEMP14 is a sequential file with C type blocks and W type records. Records contain from 66 to 135 characters.

If TEMP14 is created by the DCRPT module, it is sorted in the next processing step. The sort output is stored on TEMP15. TEMP15 must be defined as a sequential file with C type blocks and W type records. Records contain from 66 to 135 characters. The sort key begins in character position 3, is 40 characters in length, and is in display code. The sort is an ascending sort in the COBOL6 sequence.

The next processing step, which is optional, is execution of the DCIDX module. This produces an index for each Report request that specified this optional index of Catalogue names. TEMP15 is input to the DCIDX module.

Operational Considerations

The minimum field length requirements for the Report function are as follows:

DCSEL	50000 ₈ words
DCRPT	45000 ₈ words
DCIDX	17000 ₈ words

The programs that constitute the DCSEL and DCRPT modules are overlaid during processing. The DCIDX module contains one program.

Up to 99 Report requests can be batched together for processing efficiency. The result of the Report requests is an extracted file (TEMP12) that must be passed to the sort operation. The output of the sort (TEMP13) is passed to the report writer (the DCRPT module), which writes all the reports. If indexes are requested, the report writer creates a temporary file (TEMP14) that must be sorted before the index print program (the DCIDX module) is executed.

After sorting TEMP12, which is created by the DCSEL module, the DCRPT module must be executed. After sorting TEMP14, the DCIDX module must be executed.

Run time options are selected for requested reports by the OPTIONS statement. Each report requested has its own OPTIONS statement.

Output

Printed output from the Report function includes several control reports in addition to the requested reports. The conclusion of processing by the DCSEL module is indicated by the following printed totals:

```
*** TOTAL REPORT REQUESTS ATTEMPTED = n
*** TOTAL REQUESTS ACCEPTED = n
*** TOTAL REQUESTS REJECTED = n
```

These totals are followed by the Extract Control Report, which provides selection control totals for each request and for the run.

The DCRPT module produces a control report for each request and a control report for the run. The last page is titled RUN TOTALS.

The DCIDX module produces an index for each request that selected the index option.

GENERATE FUNCTION

Source statements are generated from definitions stored in the Catalogue through the Generate function. COBOL data definitions, PL/I structures, and TOTAL DBDLs can be produced. The Generate function is described in detail in section 7.

Required Files

The Generate function is implemented by execution of the programs in the DCRET module. All three files in the Catalogue (MAST1, MAST2, and MAST3) are required. Card image input transactions are also required.

The Generate function outputs the generation report to the system file OUTPUT and the generated statements to the file PNCHFIL. PNCHFIL is a sequential file with C type blocks and Z type records. Figure 10-7 shows the file usage flow for the Generate function. The control statements required to execute the Generate function are shown in figure 10-8.

Operational Considerations

The minimum field length requirement for the Generate function is 60000g words. The programs that constitute the function are overlaid during processing.

NOS Operating System

```

Job statement
USER control statement
CHARGE control statement
ATTACH,DCRET/UN=LIBRARY.
ATTACH,MAST1,MAST2,MAST3/NA.
DEFINE,PNCHFIL.
FILE,MAST1,FO=DA,EFC=3.
FILE,MAST2,FO=DA,EFC=3.
FILE,MAST3,FO=WA,EFC=3.
DCRET.
REWIND,PNCHFIL.
COPYSBF,PNCHFIL,OUTPUT.
PURGE,PNCHFIL.
CRMEP,LO.
    
```

NOS/BE Operating System

```

Job statement
ATTACH,DCRET,ID=LIBRARY.
ATTACH,MAST1,ID=name.
ATTACH,MAST2,ID=name.
ATTACH,MAST3,ID=name.
REQUEST,PNCHFIL,*PF.
FILE,MAST1,FO=DA,EFC=3.
FILE,MAST2,FO=DA,EFC=3.
FILE,MAST3,FO=WA,EFC=3.
DCRET.
REWIND,PNCHFIL.
COPYSBF,PNCHFIL,OUTPUT.
PURGE,PNCHFIL.
CRMEP,LO.
    
```

Figure 10-8. Generate Function Control Statements

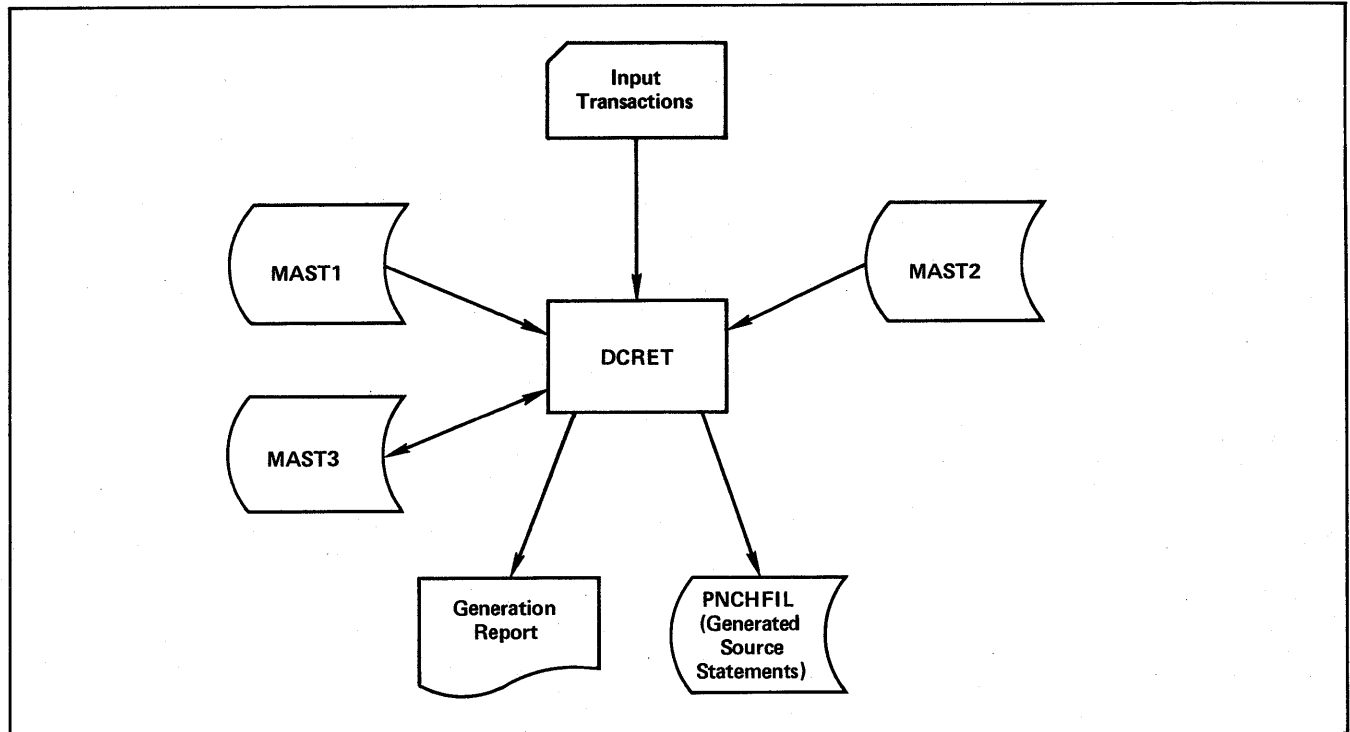


Figure 10-7. Generate Function File Usage

Pseudo-switch SW1 must be set to ON before the DCRET module is called into execution for interactive mode of operation. For batch mode of operation, switch SW1 must be set to OFF before the DCRET module is called into execution.

The Generate and Query functions both require execution of the DCRET module; therefore, these functions can be combined in one call for execution of DCRET. The functions can be requested in any order. No sequencing by key fields is required.

Run time options for a Generate request are selected by the OPTIONS statement. One statement can be specified for each request.

Output

The Generate function produces language-dependent source statements for COBOL and PL/I as well as TOTAL DBDLs. The output can be listed, written to a sequential file, or both listed and written. Suppression of the listing or the sequential file is a run time option.

The final output for a Generate request is as follows:

```
*** END GENERATION ***
```

After all Generate and Query requests have been performed, the following message is output:

```
*** END DATA CATALOGUE RETRIEVAL REPORTS ***
```

CONVERT FUNCTION

The Convert function scans existing COBOL programs and TOTAL DBDLs and then creates input transactions for the Update function. An audit trail of the data structures processed is also produced. The Convert function is described in detail in section 8.

Required Files

The Convert function is implemented in three steps. The DCCONVT module is executed to scan the target definitions and extract a work file. The work file is sorted and the DCCONGN module is executed to create input transactions for the Update function. Only the control file (MAST3) in the Catalogue is required for the Convert function. The control statements required to execute the Convert function are shown in figure 10-9. Figure 10-10 shows the file usage flow for the Convert function.

The COBOL programs or TOTAL DBDLs to be scanned are read from the file CONVFIL, which can reside on a system internal formatted tape or disk. CONVFIL is a sequential file with C type blocks and Z type records.

Two temporary files are used by the Convert function. The DCCONVT module creates the temporary file WRKFILE, which contains data extracted from CONVFIL. This work file is then sorted and the sort output is stored on the temporary file TEMP1. WRKFILE and TEMP1 are sequential files with C type blocks and W type records; records are from 120 to 258 characters in length. Both files can reside on either a system internal formatted tape or disk.

NOS Operating System

```
Job statement
USER control statement
CHARGE control statement
ATTACH,MAST3/NA.
ATTACH,DCCONVT,DCCONGN/UN=LIBRARY.
GET,CONVFIL.
FILE,WRKFILE,BT=C,RT=W,MNR=120,MRL=258,CM=NO.
FILE,TEMP1,BT=C,RT=W,MNR=120,MRL=258,CM=NO.
DCCONVT.
SORTMRG(0=VMSG/R)
DCCONGN.
REWIND,PNCHFIL.
COPYSBF,PNCHFIL,OUTPUT.
```

NOS/BE Operating System

```
Job statement
ATTACH,MAST3,ID=name.
ATTACH,DCCONVT,ID=LIBRARY.
ATTACH,DCCONGN,ID=LIBRARY.
GET,CONVFIL.
FILE,WRKFILE,BT=C,RT=W,MNR=120,MRL=258,CM=NO.
FILE,TEMP1,BT=C,RT=W,MNR=120,MRL=258,CM=NO.
DCCONVT.
SORTMRG(0=VMSG/R)
DCCONGN.
REWIND,PNCHFIL.
COPYSBF,PNCHFIL,OUTPUT.
```

Figure 10-9. Convert Function Control Statements

WRKFILE is input to the sort operation. The sort key begins in character position 1, is 75 characters in length, and is in display code. The sort is an ascending sort in the COBOL6 sequence.

The final output of the Convert function is written on the file PNCHFIL. This file contains input transactions for the Update function. PNCHFIL can reside on a system internal formatted tape or disk. It is a sequential file with C type blocks and Z type records.

The DCCONVT module requires input transactions, system file OUTPUT, CONVFIL, MAST3, and WRKFILE. The DCCONGN module requires system file OUTPUT, TEMP1, and PNCHFIL.

Operational Considerations

The minimum field length requirements for the Convert function are as follows:

```
DCCONVT 430008 words
```

```
DCCONGN 220008 words
```

The programs are overlaid by several phases during processing.

Statements in the Convert request must be entered as defined in section 8. No sequencing by key fields is required.

The Convert function has no run time options governing operational considerations.

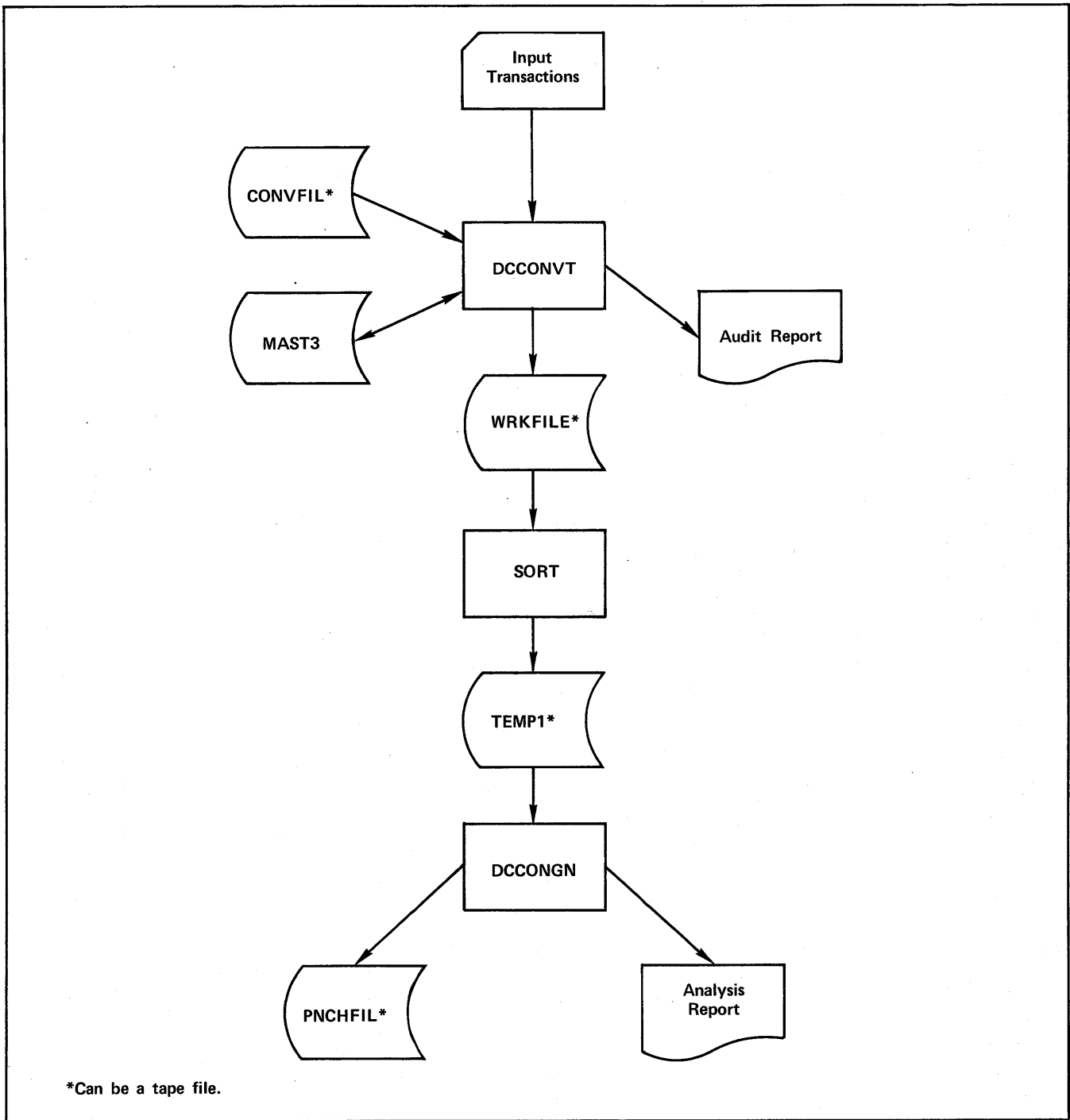


Figure 10-10. Convert Function File Usage

Output

Output from the first phase of the Convert function includes an audit trail of the data structure scanned and error messages produced by the DCCONVT module. The second phase output includes an analysis report and a file containing the generated Update function transactions.

The final output for the Convert function is the following message:

*** END OF CONVERSION CONTROL REPORT ***

Several examples are presented in this section to illustrate the use of the various Data Catalogue functions. Samples of input statements and output reports are shown for each function.

UPDATE FUNCTION

The Update function is used to maintain the Catalogue. New entries can be added, existing entries can be changed, and existing entries or portions of entries can be deleted. The function produces a report of transactions applied to the Catalogue and shows errors that were detected in editing transactions.

Figure 11-1 illustrates the control statements and the input statements for an Update request. The three Catalogue files and the module DCUPD must be attached. FILE control statements are required for the Catalogue files if the CRM error file messages and statistics are desired; the CRMEP control statement is required to print the error file. The function input statements shown in the illustration are a partial listing of the actual statements used in the run. These statements add three entries to the Catalogue: an element entity (PART-NO), a record entity (PART-REC), and a module entity (CREATE).

The Update Audit Report, which is output by the Update function, is shown in figure 11-2. This report shows the OPTIONS statement values in effect and then lists the input transactions. The final portion of the Update Audit Report lists statistical totals for the complete Update request.

QUERY FUNCTION

The Query function is used to interrogate the Catalogue files for information based on the contents of fields in entries, on relationships between entries, or on combinations of these characteristics. Queries can be simple or complex; responses can be summarized or detailed.

The control statements and the input statements for a Query request are shown in figure 11-3. The module DCRET and the three Catalogue files must be attached. FILE control statements are required for the three Catalogue files if the CRM error file messages and statistics are desired; the CRMEP control statement is required to print the error file. The input statements include twelve queries.

Figure 11-4 shows a portion of the Query Response Report produced for the Query request shown in figure 11-3. The report shows the first query (COUNT ELEMENTS) and the response (12 ENTRIES QUALIFY). This is followed by the second query (LIST ELEMENTS) and its response, which indicates the number of qualifying entries and the Catalogue name and entity type of each entry. The response to the third query (SHOW ELEMENTS) consists of the number of qualified entries, the Catalogue name and entity type of each entry, and a detailed listing of each entry.

The final output from the Query function indicates the total number of queries attempted, the number accepted, and the number rejected.

REPORT FUNCTION

The Report function is used to generate various reports in predetailed presentations of the stored data in the Catalogue. Reports that are primarily for finished documentation can be printed in 8-1/2 by 11 format rather than in standard computer-width format. Reports used for analysis are printed in computer-width format.

Figure 11-5 shows the control statements and input for a typical Report function run. The three Catalogue files and the modules DCRPT, DCSEL, and DCIDX must be attached. FILE control statements are required for the Catalogue files if the CRM error file messages and statistics are desired; the CRMEP control statement is required to print the error file. FILE control statements are also required for the local files TEMP12, TEMP13, TEMP14, and TEMP15.

Several Report requests are included in the input statements for the Report function. These statements are followed by two sets of directives for the sort operations.

The Report function outputs two special reports before the requested reports are produced. The first report, the Report Request Audit, is illustrated in figure 11-6. This report provides an audit trail of the Report requests input to the function. At the end of the report, totals are listed for the number of requests attempted, the number accepted, and the number rejected.

The Extract Control Report, illustrated in figure 11-7, lists each Report request and indicates the number of records extracted for the requested report. The total number of requests and the total number of records extracted by the run are listed at the end of the Extract Control Report.

CATALOGUE REPORT

Figure 11-8 illustrates the type of output generated by the Report function for a Catalogue Report request. The Request Summary Report shows the Catalogue Report was requested and that 27 Catalogue entries were selected for the report. The status of the output options is also shown.

A detailed listing of each selected entry follows the Request Summary Report. Figure 11-8 shows the listing for one Catalogue entry, BACK-ORDER.

When all entries have been listed, the Request Control Report provides control totals for the request. The report in figure 11-8 shows the number of entries for each entity type; a total of 27 entries were selected from the Catalogue for the report. A total of 17 records were written on the index file.

INDEX REPORT

The output generated by the Report function for an Index Report request is shown in figure 11-9. The Request Summary Report shows a request for an Index Report by Catalogue name and indicates that 27 entries were selected for the report. The status of the output options is also shown. This is followed by the Index Report generated for the selected entries.

RELATIONAL REPORT

Figure 11-10 illustrates the output generated by the Report function for a Relational Report request. The Request Summary Report shows a request for the Relational Report by Catalogue name and indicates that 195 entries were selected for the report. The status of the output options is also shown. A portion of the report produced as a result of the request is shown in figure 11-10.

NAME ANALYSIS REPORT

Figure 11-11 illustrates the output generated by the Report function for a Name Analysis Report request. The Request Summary Report shows a request for the Name Analysis Report by Catalogue name and indicates that 27 entries were selected for the report. The status of the output options is also shown. The report generated by this request is also shown in figure 11-11.

HIERARCHY REPORT

The output generated by the Report function for a Hierarchy Report is illustrated in figure 11-12. The Request Summary Report shows that the Hierarchy Report was requested for IN-STOCK-REPORT and that 13 Catalogue entries were selected for the report. The status of the output options is also shown.

The Hierarchy Report for IN-STOCK-REPORT is in ascending hierarchical sequence; the output option SEQ defaults to TYPE, which indicates the ascending sequence. As shown in figure 11-12, the report begins with the element entity (BACK-ORDER) and proceeds up the hierarchy to the report entity (IN-STOCK-REPORT).

The final output is the Request Control Report, which provides control totals for the request. In figure 11-12, this report shows that 11 element entities, 1 record entity, and 1 report entity were selected and that a total of 18 records were written on the index file.

USAGE REPORT

Figure 11-13 illustrates the output generated by the Report function for a Usage Report request. The Request Summary Report shows that the Usage Report was requested for PROFIT-PROG and that two Catalogue entries were selected for the report. The status of output options is also shown.

The Usage Report for PROFIT-PROG is in ascending hierarchical sequence; the output option SEQ defaults to TYPE, which indicates the ascending sequence. As shown

in figure 11-13, the report begins with the module entity (PROFIT-PROG) and proceeds up the hierarchy to the system entity (INTER-ACT).

The final output is the Request Control Report, which provides control totals for the Usage Report. In figure 11-13, one module entity and one system entity were selected for the report and six records were written on the index file.

FUNCTION REPORTS

When all Report requests have been processed, the Report function produces a Request Control Report that provides control totals for all requests input to the function. This report is shown in figure 11-14. Indexes are then produced for each request that selected the INDEX=YES option. A sample index is shown in figure 11-15.

GENERATE FUNCTION

The Generate function is used to produce source statements from entries stored in the Catalogue. COBOL data definitions, PL/I structures, and TOTAL DBDLs can be generated.

Figure 11-16 shows the control statements and the function input statements for generating COBOL data definitions. The three Catalogue files and the module DCRET must be attached. FILE control statements are required for the Catalogue files if the CRM error file messages and statistics are desired; the CRMEP control statement is required to print the error file.

The output produced by the Generate function is shown in figure 11-17. The Generation Report lists the input statements and then lists the data definitions generated by the function. The final output provides totals for the number of commands attempted, the number accepted, and the number rejected.

CONVERT FUNCTION

The Convert function scans existing COBOL programs and TOTAL DBDLs to create input transactions for adding entries to the Catalogue. An optional renaming facility allows new names to be assigned to selected entries.

Figure 11-18 shows the control statements, function input statements, and sort directives for a COBOL Convert request. The MAST3 Catalogue file and the modules DCCONVT and DCCONGN must be attached. FILE control statements are required for two local files (WRKFILE and TEMP1).

The output generated by the Convert function is shown in figure 11-19. The Conversion Edit Report lists the input statements for the request and provides totals indicating the number of modules accepted, the number rejected, and the number read. The Cobol Conversion Report shows the data structure selected for conversion. The final output, the Conversion Control Report, lists the Update function input transactions created by the Convert function.

NOS Operating System

```

Job statement.
USER control statement.
CHARGE control statement.
ATTACH,DCUPD/UN=LIBRARY.
ATTACH,MAST1,MAST2,MAST3/M=W,NA.
FILE,MAST1,FO=DA,EFC=3.
FILE,MAST2,FO=DA,EFC=3.
FILE,MAST3,FO=WA,EFC=3.
DCUPD.
CRMEP,LO.
EXIT.
DMP.
DMP,377777.
CRMEP,LO.
7/8/9 card

```

NOS/BE Operating System

```

Job statement.
ATTACH,DCUPD.
ATTACH,MAST1,IO=DC2,PW=DC2.
ATTACH,MAST2,IO=DC2,PW=DC2.
ATTACH,MAST3,IO=DC2,PW=DC2.
FILE,MAST1,FU=DA,EFC=3.
FILE,MAST2,FU=DA,EFC=3.
FILE,MAST3,FU=WA,EFC=3.
DCUPD.
EXTEND(MAST1)
CRMEP,LU.
EXIT.
DMP.
DMP,377777.
CRMEP,LO.
7/8/9 card

```

```

$UPDATE
OPTICNS REV=NO=NC-CHK,USER=STH
ADD ELE=PART-NO
CONTROL
1 ESTATUS=E
DESCRIPTION
1 THIS IS A PARTICULAR PARTS ID NUMBER
CRIGIN
1 DEPT=PURCHASING,PROGRAM=PTADD,FORM=FNWPT,GENCODE=N
NAMES
1 DATANAME=PART-NO
ATTRIBUTES
1 LENGTH=6,FORMAT=C,PICTURE=X(6),VOLATILE=S
ADD REC=PART-REC
DES
1 THIS RECORD HAS ALL INFORMATION ABOUT PARTS
5 WHICH THE FIRM HAS IN ITS INVENTORY.
CLASSIFICATION
1 PART,PARTS,INVENTORY,STOCK
NAM
1 CATANAME=PART-REC
ATT
1 MAXLEN=80,MINLEN=80
STR
1 CATNAME=PART-NC,KEY=U
5 CATNAME=NC-PART
10 CATNAME=BACK-ORDER,KEY=C
:
:
:
ADD MOD=CREATE
DES
1 THIS PROGRAM INITIALIZES THE DATA BASE
RES
1 STATUS=C,FUNCTION=D,PERSON=CREATE PROGRAMMER
NAM
1 NAME=CREATE
ATT
1 LANGPRG=C,TYPEPRG=B,CVLYPRG=N
STORAGE
1 STYPE=C,PLACE=P.F.,MEDIUM=DISK,SIZE=20000,UNITSIZE=3
OPERATING
1 REGION=20,TELAPSED=1,UELAPSED=S,TIMECPU=1,UNITCPU=1,NAMEPROC=PROFPRCC
SCHEDULING
1 NBEGIN=0800,NEND=1700,NDAY=MTWTF
RELATIONAL
1 CATNAME=PART-REC
5 CATNAME=OPN-FLS
10 CATNAME=RD-INP

```

Figure 11-1. Update Request Example

DATA CATALOGUE 2
UPDATE AUDIT REPORT

OPTIONS REV=NO=NO=CHK,USER=STH

OPTIONS IN EFFECT THIS UPDATE :
REVISION NUMBER=NO CHECKING
EDIT ONLY =NO
QUOTE =QUOTE
USER =STH

*
* INPUT TO ADD BUSINESS WORLD DB ENTRIES
*

*
ADD ELE=PART-NO
*

CCNTRCL
1 ESTATUS=E
DESCRIPTION
1 THIS IS A PARTICULAR PARTS ID NUMBER
ORIGIN
1 DEPT=PURCHASING,PROGRAM=PTADD,FOR=FWMPT,GENCODE=N
NAMES
1 DATANAME=PART-NO
ATTRIBUTES
1 LENGTH=6,FORMAT=C,PICTURE=X(6),VOLATILE=S

*
ADD REC=PART-REC
*

DES
1 THIS RECCRD HAS ALL INFORMATION ABOUT PARTS
5 WHICH THE FIRM HAS IN ITS INVENTORY.
CON
1 ESTATUS=E

⋮

UPDATE AUDIT REPORT

UPDATE STATISTICS THIS RUN:
TOTAL TX IMAGES READ 336
TOTAL ENTRIES PROCESSED 27
TOTAL ENTRIES ADDED 27
TOTAL ENTRIES CHANGED 2
TOTAL ENTRIES DELETED 0
TOTAL SERIOUS ERRORS 0
TOTAL WARNINGS 0
TOTAL ELEMENT 12
TOTAL RECORD 2
TOTAL FILE 1
TOTAL FORM 1
TOTAL REPORT 2
TOTAL MODULE 8
TOTAL SYSTEM 1
*** END AUDIT REPORT ***

Figure 11-2. Update Audit Report

NOS Operating System

Job statement.
USER control statement.
CHARGE control statement.
ATTACH,DCRET/UN=LIBRARY.
ATTACH,MAST1,MAST2,MAST3/NA.
FILE,MAST1,F0=DA,EFC=3.
FILE,MAST2,F0=DA,EFC=3.
FILE,MAST3,F0=WA,EFC=3.
DCRET.
CRMEP,LO.
EXIT.
DMP.
DMP,377777.
CRMEP,LO.
7/8/9 card

NOS/BE Operating System

Job statement.
ATTACH,MAST1,ID=DC2,PW=DC2.
ATTACH,MAST2,ID=DC2,PW=DC2.
ATTACH,MAST3,ID=DC2,PW=DC2.
ATTACH,DCRET.
FILE,MAST1,FU=DA,EFC=3.
FILE,MAST2,FU=DA,EFC=3.
FILE,MAST3,FU=WA,EFC=3.
DCRET.
CRMEP,LO.
EXIT.
DMP.
DMP,377777.
CRMEP,LO.
7/8/9 card

\$QUERY
TITLE #QUERY TITLE#
COUNT ELEMENTS
LIST ELEMENTS
SHOW ELEMENTS
LIST ELEMENTS WITH LENGTH = 6
LIST ELEMENTS WITH LENGTH > 10
LIST ELEMENTS WITH LENGTH = 10 AND FORMAT = F
LIST REC WHICH-USE PART-NO
LIST ELE WITH DES CS #AMOUNT#
LIST ELE HAVING ORIGIN MISSING
LIST FIL TO FOR
SHOW ENTRIES WITH KW EQ #PART#
LIST #A# TO #F#
6/7/8/9 card

Figure 11-3. Query Request Example

DATA CATALOGUE 2
QUERY RESPONSE REPORT

QUERY TITLE

\$QUERY
COUNT ELEMENTS

QUERY RESPONSE REPORT
QUERY TITLE

12 ENTRIES QUALIFY.

QUERY RESPONSE REPORT
QUERY TITLE

LIST ELEMENTS

QUERY RESPONSE REPORT
QUERY TITLE

12 ENTRIES QUALIFY.

PART-NO	ELEMENT
NO-PART	ELEMENT
BACK-ORDER	ELEMENT
CN-ORDER	ELEMENT
REORDER-PT	ELEMENT
LEAD-TIME	ELEMENT
UNIT-WEIGHT	ELEMENT
UNIT-COST	ELEMENT
UNIT-PRICE	ELEMENT
DESCRIPTION	ELEMENT
CHECKSUM	ELEMENT
STATIST	ELEMENT

Figure 11-4. Query Response Report (Sheet 1 of 3)

QUERY RESPONSE REPORT
QUERY TITLE

SHOW ELEMENTS

QUERY RESPONSE REPORT
QUERY TITLE

12 ENTRIES QUALIFY.

PART-NO	ELEMENT
NC-PART	ELEMENT
BACK-ORDER	ELEMENT
CR-ORDER	ELEMENT
REORDER-PT	ELEMENT
LEAD-TIME	ELEMENT
UNIT-WEIGHT	ELEMENT
UNIT-COST	ELEMENT
UNIT-PRICE	ELEMENT
DESCRIPTION	ELEMENT
CHECKSUM	ELEMENT
STATEST	ELEMENT

Figure 11-4. Query Response Report (Sheet 2 of 3)

QUERY RESPONSE REPORT

QUERY TITLE

CATALOGUE NAME	REV	LINE	TYPE
PART-NO			ELEMENT
			---CONTROL CATEGORY---
	1	1	STATUS=EXISTING
			---DESCRIPTION CATEGORY---
	1	1	THIS IS A PARTICULAR PARTS ID NUMBER
			---ORIGIN CATEGORY---
	1	1	DEPARTMENT=PURCHASING PROGRAM=PTADD FORM=FNWPT GENERATED=NO
			---NAME CATEGORY---
	1	1	DATA NAME=PART-NO
			---ATTRIBUTE CATEGORY---
	1	1	LENGTH= 6 FORMAT=CHARACTER VOLATILITY=STATIC PICTURE=X(6)
			⋮

QUERY RESPONSE REPORT

QUERY TITLE

*** TOTAL QUERY COMMANDS ATTEMPTED= 12
 *** TOTAL COMMANDS ACCEPTED= 12
 *** TOTAL COMMANDS REJECTED= 0

QUERY RESPONSE REPORT

QUERY TITLE

*** END DATA CATALOGUE RETRIEVAL REPORTS ***

Figure 11-4. Query Response Report (Sheet 3 of 3)

NOS Operating System

NOS/BE Operating System

```

Job statement
USER control statement.
CHARGE control statement.
ATTACH,MAST1,MAST2,MAST3/NA.
FILE,MAST1,FO=DA,EFC=3.
FILE,MAST2,FO=DA,EFC=3.
FILE,MAST3,FO=WA,EFC=3.
ATTACH,DCRPT,DCSEL,DCIDX/UN=LIBRARY,NA.
FILE,TEMP12,BT=C,RT=W,MRL=294,MNR=100.
FILE,TEMP13,BT=C,RT=W,MRL=294,MNR=100.
DCSEL.
REWIND,TEMP12.
SORTMRG(O=VMSG/R)
FILE,TEMP14,BT=C,RT=W,MRL=135,MNR=66,CM=NO.
FILE,TEMP15,BT=C,RT=W,MRL=135,MNR=66,CM=NO.
DCRPT.
REWIND,TEMP14.
SORTMRG(O=VMSG/R)
DCIDX.
CRNEP,LO.
EXIT.
DMP.
DMP,377777.
CRNEP,LO.
7/8/9 card

```

```

Job statement.
ATTACH,MAST1,ID=OC2,PW=DC2.
ATTACH,MAST2,ID=OC2,PW=DC2.
ATTACH,MAST3,ID=OC2,PW=DC2.
ATTACH,DCSEL.
ATTACH,DCRPT.
ATTACH,DCIDX.
FILE,MAST1,FO=DA,EFC=3.
FILE,MAST2,FO=DA,EFC=3.
FILE,MAST3,FO=WA,EFC=3.
FILE,TEMP12,BT=C,RT=W,MRL=294,MNR=100.
FILE,TEMP13,BT=C,RT=W,MRL=294,MNR=100.
DCSEL.
REWIND,TEMP12.
SORTMRG(O=VMSG/R)
FILE,TEMP14,BT=C,RT=W,MRL=135,MNR=66,CM=NO.
FILE,TEMP15,BT=C,RT=W,MRL=135,MNR=66,CM=NO.
DCRPT.
REWIND,TEMP14.
SORTMRG(O=VMSG/R)
DCIDX.
CRNEP,LO.
EXIT.
DMP.
DMP,377777.
CRNEP,LO.
7/8/9 card

```

```

$REPORT CATALOGUE
TITLE #CLASS CATALOGUE REPORT 1#
$REPORT CATALOGUE
TITLE #CLASS CATALOGUE REPORT 2#
OPTIONS FORMAT=DOCUM,SEC=REVERSE
$REPORT INDEX BY CATNAME
TITLE #CLASS INDEX REPORT 1#
$REPORT INDEX BY DATANAME
TITLE #CLASS INDEX REPORT 2#
OPTIONS SEQ=REVERSE,MESSAGE=YES
$REPORT INDEX BY NAME
$REPORT RELATIONAL BY CATNAME
TITLE #CLASS RELATIONAL REPORT 1#
OPTIONS INDIRECT=YES
$REPORT NAME-ANALYSIS BY CATNAME
$REPORT USAGE OF IN-STOCK-REPORT
TITLE #CLASS USAGE REPORT 1#
OPTIONS INDENTED-INDEX=YES,MESSAGE=YES
$REPORT HIERARCHY OF IN-STOCK-REPORT
TITLE #CLASS HIERARCHY REPORT 1#
$REPORT USAGE OF PROFIT-PROG
$REPORT HIERARCHY OF PROFIT-PROG
7/8/9 card

```

```

SORT
FILE,INPUT=TEMP12(R),OUTPUT=TEMP13(CR)
FIELD,SRTKEY(3,75,DISPLAY)
KEY,SRTKEY(A,C080L6)
END
7/8/9 card

```

```

SORT
FILE,INPUT=TEMP14(R),OUTPUT=TEMP15(CR)
FIELD,SRTKEY(3,40,DISPLAY)
KEY,SRTKEY(A,C080L6)
END
6/7/8/9 card

```

Figure 11-5. Report Request Example

DATA CATALOGUE 2
THE DATA CATALOGUE MANUAL

DATA CATALOGUE 2

REPORT REQUEST AUDIT

CLASS CATALOGUE REPORT 1

\$REPORT CATALOGUE

*** REPORT REQUEST COMPLETE ***

REPORT REQUEST AUDIT

CLASS CATALOGUE REPORT 2

\$REPORT CATALOGUE
OPTIONS FORMAT=DOCUM, SEQ=REVERSE

*** REPORT REQUEST COMPLETE ***

⋮

REPORT REQUEST AUDIT

*** TOTAL REPORT REQUESTS ATTEMPTED= 11

*** TOTAL COMMANDS ACCEPTED= 11

*** TOTAL COMMANDS REJECTED= 0

REPORT REQUEST AUDIT

*** END DATA CATALOGUE RETRIEVAL REPORTS ***

Figure 11-6. Report Request Audit

DATA CATALOGUE 2
THE DATA CATALOGUE MANUAL

DATA CATALOGUE 2
EXTRACT CONTROL REPORT

REPORT/FILE REQUEST NUMBER 1

\$REPORT CATALOGUE
TITLE =CLASS CATALOGUE REPORT 1
** 27 RECCRS WERE EXTRACTED BY THIS REQUEST

REPORT/FILE REQUEST NUMBER 2

\$REPORT CATALOGUE
TITLE =CLASS CATALOGUE REPORT 2
** 27 RECCRS WERE EXTRACTED BY THIS REQUEST

⋮

EXTRACT CONTROL REPORT

*** 11 REPORT/FILE REQUESTS WERE PROCESSED
*** 352 RECORDS WERE EXTRACTED BY THIS RUN

EXTRACT CONTROL REPORT

*** END EXTRACT CONTROL REPORT ***

Figure 11-7. Extract Control Report

DATA CATALOGUE 2
REQUEST SUMMARY REPORT

REPORT/FILE REQUEST NUMBER 1

*** THE FOLLOWING PARAMETERS WERE SPECIFIED-

\$REPORT CATALOGUE

*** THE STATUS OF APPLICABLE OPTICNS IS SHOWN BELOW-

SEQ = TYPE
INDEX = YES
NEWPAGE = YES
FORMAT = STANDARD
SHOWREQ = YES
MESSAGE = NO

*** ENTRY SELECTION STATUS IS SUMMARIZED BELOW-

** 27 ENTRIES WERE SELECTED BY THIS REQUEST

CATALOGUE REPORT
CLASS CATALOGUE REPORT 1

CATALOGUE NAME	REV	LINE	TYPE
BACK-ORDER			ELEMENT
			---CONTROL CATEGORY---
	1	1	STATUS=EXISTING
			---DESCRIPTION CATEGORY---
	1	1	1 THE NUMBER OF THESE PARTS CURRENTLY
	1	5	5 BACKORDERED TO CUSTOMERS.
			---ORIGIN CATEGORY---
	1	1	1 DEPARTMENT=PURCHASING PRCGRP=PROORD CYCLE=DAILY GENERATED=YES
			---NAME CATEGORY---
	1	1	1 DATA NAME=BACK-ORDER
			---ATTRIBUTE CATEGORY---
	1	1	1 LENGTH= 6 FORMAT=NUMERIC VOLATILITY=DYNAMIC NATURE=QUANTIT PICTURE=9(6)
			⋮

Figure 11-8. Catalogue Report (Sheet 1 of 2)

REQUEST CONTROL REPORT
CLASS CATALOGUE REPORT 1

REPORT/FILE REQUEST NUMBER 1

*** ENTRY TYPE CONTROL TOTALS

TYPE OF ENTRY	NUMBER OF		ENTRIES WRITTEN ON \$FILE	BYPASSED (DUPLICATES)
	READ	PRINTED IN \$REPORT		
ELEMENT	12	12	0	0
RECORD	2	2	0	0
FILE	1	1	0	0
FORM	1	1	0	0
REPORT	2	2	0	0
MODULE	8	8	0	0
SYSTEM	1	1	0	0
** REQUEST TOTALS	27	27	0	0

*** FILE RECORD COUNT CONTROL TOTALS

** 37 RECORDS WERE WRITTEN ON THE INDEX FILE
** 0 RECORDS WERE WRITTEN ON THE USER FILE (\$FILE)

***** END CONTROL REPORT FOR REQUEST 1 *****

Figure 11-8. Catalogue Report (Sheet 2 of 2)

DATA CATALOGUE 2
REQUEST SUMMARY REPORT

REPORT/FILE REQUEST NUMBER 3

*** THE FOLLOWING PARAMETERS WERE SPECIFIED-

\$REPORT INDEX BY CATNAME

*** THE STATUS OF APPLICABLE OPTIONS IS SHOWN BELOW-

REPORT = YES
SEQ = TYPE
FORMAT = STANDARD
SHOWREQ = YES
MESSAGE = NO

*** ENTRY SELECTION STATUS IS SUMMARIZED BELOW-

** 27 ENTRIES WERE SELECTED BY THIS REQUEST

INDEX REPORT BY CATNAME

ENTRY TYPE	CATALOGUE NAME
BACK-ORDER	BACK-ORDER
CHECKSUM	CHECKSUM
DESCRIPTION	DESCRIPTION
LEAD-TIME	LEAD-TIME
NO-PART	NO-PART
ON-ORDER	ON-ORDER
PART-NO	PART-NO
REORDER-PT	REORDER-PT
STATST	STATST
UNIT-COST	UNIT-COST
UNIT-PRICE	UNIT-PRICE
UNIT-WEIGHT	UNIT-WEIGHT
PART-REC	PART-REC
PROFIT-REC	PROFIT-REC
INVENTORY	INVENTORY
FORM-A3	FORM-A3
IN-STOCK-REPORT	IN-STOCK-REPORT
PROFIT-SCREEN	PROFIT-SCREEN
COMPUTE-PROFIT	COMPUTE-PROFIT
CREATE	CREATE
FINISH	FINISH
OPN-FLS	OPN-FLS
PART-ERR	PART-ERR
PROFIT-PROG	PROFIT-PROG
PT1	PT1
RD-INP	RD-INP
INTER-ACT	INTER-ACT

*** END OF REPORT ***

Figure 11-9. Index Report

REPORT/FILE REQUEST NUMBER 6

*** THE FOLLOWING PARAMETERS WERE SPECIFIED-

\$REPORT RELATIONAL BY CATNAME
OPTIONS INDIRECT=YES

*** THE STATUS OF APPLICABLE OPTICNS IS SHOWN BELOW-

REPORT = YES
SEQ = TYPE
FORMAT = STANDARD
SHOWREQ = YES
MESSAGE = NO
INDIRECT = YES

*** ENTRY SELECTION STATUS IS SUMMARIZED BELOW-

** 195 ENTRIES WERE SELECTED BY THIS REQUEST

RELATIONAL REPORT BY CATNAME

RELATED ENTRIES BY TYPE

----- ELEMENT -----			
CATALOGUE NAME		CATALOGUE NAME	REMARK
BACK-ORDER	RECORD	PART-REC	
	-----	-----	
	FILE	INVENTORY	
	-----	-----	
CHECKSUM	REPORT	IN-STOCK-REPORT	
	-----	-----	
	MODLLE	CREATE	
	-----	-----	
DESCRIPTION	RECORD	PART-REC	
	-----	-----	
	FILE	INVENTORY	
	-----	-----	
	REPORT	IN-STOCK-REPORT	
	-----	-----	
		⋮	

Figure 11-10. Relational Report

DATA CATALOGUE 2
REQUEST SUMMARY REPORT

REPORT/FILE REQUEST NUMBER 7

*** THE FOLLOWING PARAMETERS WERE SPECIFIED-

\$REPORT NAME-ANALYSIS BY CATNAME

*** THE STATUS OF APPLICABLE OPTIONS IS SHOWN BELOW-

REPORT = YES
SEQ = TYPE
FORMAT = STANDARD
SHOWREQ = YES
MESSAGE = NO
BREAK CHARACTER = -
DESCRIPTION = NONE SPECIFIED

*** ENTRY SELECTION STATUS IS SUMMARIZED BELOW-

** 27 ENTRIES WERE SELECTED BY THIS REQUEST

NAME ANALYSIS BY CATNAME

NAME COMPONENT	FULL CONTEXT	ENTRY TYPE
EACK	EACK-ORDER	ELEMENT
CHECKSUM	CHECKSUM	ELEMENT
COST	UNIT-COST	ELEMENT
DESCRIPTION	DESCRIPTION	ELEMENT
LEAD	LEAD-TIME	ELEMENT
NO	NC-PART	ELEMENT
NO	PART-NO	ELEMENT
ON	CN-CRDR	ELEMENT
ORDER	EACK-ORDER	ELEMENT
ORDER	CN-CRDR	ELEMENT
PART	NC-PART	ELEMENT
PART	PART-NO	ELEMENT
PRICE	UNIT-PRICE	ELEMENT
PT	REORDER-PT	ELEMENT
REORDER	REORDER-PT	ELEMENT
STATEST	STATEST	ELEMENT
TIME	LEAD-TIME	ELEMENT
UNIT	UNIT-COST	ELEMENT
UNIT	UNIT-PRICE	ELEMENT
UNIT	UNIT-WEIGHT	ELEMENT
WEIGHT	UNIT-WEIGHT	ELEMENT
PART	PART-REC	RECORD
PROFIT	PROFIT-REC	RECORD
REC	PART-REC	RECORD
REC	PROFIT-REC	RECORD
INVENTORY	INVENTORY	FILE

⋮

Figure 11-11. Name Analysis Report

DATA CATALOGUE 2
REQUEST SUMMARY REPORT

REPORT/FILE REQUEST NUMBER 9

*** THE FOLLOWING PARAMETERS WERE SPECIFIED-

\$REPORT HIERARCHY OF IN-STOCK-REPORT

*** THE STATUS OF APPLICABLE OPTIONS IS SHOWN BELOW-

SEQ	=	TYPE
INDEX	=	YES
NEWPAGE	=	YES
FORMAT	=	STANDARD
SHOWREQ	=	YES
MESSAGE	=	NO
INDENTED-INDEX	=	NO
STOPENT	=	NONE SPECIFIED

*** ENTRY SELECTION STATUS IS SUMMARIZED BELOW-

** 13 ENTRIES WERE SELECTED BY THIS REQUEST

HIERARCHY OF IN-STOCK-REPORT
CLASS HIERARCHY REPORT 1

CATALOGUE NAME	REV	LINE	TYPE	
BACK-ORDER			ELEMENT	
			--- <td></td>	
	1	1		STATUS=EXISTING
			---DESCRIPTION CATEGORY---	
	1	1	1 THE NUMBER OF THESE PARTS CURRENTLY	
	1	5	5 BACKORDERED TO CUSTOMERS.	
			---ORIGIN CATEGORY---	
	1	1	1 DEPARTMENT=PURCHASING	CYCLE=DAILY
			PRCGRP=PROORC	GENERATED=YES
			---NAME CATEGORY---	
	1	1	1 DATA NAME=BACK-ORDER	
			---ATTRIBUTE CATEGORY---	
	1	1	1 LENGTH= 6 FORMAT=NUMERIC	
			VCLATILITY=DYNAMIC NATURE=QUANTIT	
			PICTURE=9(6)	
			:	
			:	

Figure 11-12. Hierarchy Report (Sheet 1 of 3)

HIERARCHY OF IN-STOCK-REPORT

CLASS HIERARCHY REPORT 1

CATALOGUE NAME	REV	LINE	TYPE	
PART-REC			RECORD	
---CONTROL CATEGORY---				
	1	1		STATUS=EXISTING
---CLASSIFICATION CATEGORY---				
	1	1	PART,PARTS,INVENTORY,STOCK	
---DESCRIPTION CATEGORY---				
	1	1	THIS RECORD HAS ALL INFORMATION ABOUT PARTS	
	1	5	WHICH THE FIRM HAS IN ITS INVENTORY.	
---NAMES CATEGORY---				
	1	1	DATA NAME=PART-REC	
---ATTRIBUTE CATEGORY---				
	1	1	MAXIMUM LENGTH= 80 MINIMUM LENGTH= 80	
---STRUCTURE CATEGORY---				
			CATALOGUE NAME	ALIAS FILLER KEY
	1	1	PART-NC	UNIQUE
	1	5	NC-PART	
	1	10	BACK-ORDER	DUPLICATE
	1	15	ON-ORDER	
	1	20	REORDER-PT	
	1	25	LEAD-TIME	
	1	30	UNIT-WEIGHT	
	1	35	UNIT-COST	
	1	40	UNIT-PRICE	
	1	45	DESCRIPTION	
	1	50	CHECKSLM	

Figure 11-12. Hierarchy Report (Sheet 2 of 3)

REQUEST SUMMARY REPORT

REPORT/FILE REQUEST NUMBER 10

*** THE FOLLOWING PARAMETERS WERE SPECIFIED-

REPORT USAGE OF PROFIT-PROG

*** THE STATUS OF APPLICABLE OPTIONS IS SHOWN BELOW-

SEQ = TYPE
INDEX = YES
NEWPAGE = YES
FORMAT = STANDARD
SHOWREQ = YES
MESSAGE = NO
INDENTED-INDEX = NO
STOFENT = NONE SPECIFIED

*** ENTRY SELECTION STATUS IS SUMMARIZED BELOW-

** 2 ENTRIES WERE SELECTED BY THIS REQUEST

USAGE OF PROFIT-PROG

CATALOGUE NAME REV LINE TYPE

PROFIT-PROG

MOCCLE

---DESCRIPTION CATEGORY---

1 1 THIS PROGRAM RUNS IN BATCH MODE AND OUTPUTS THE PROFIT
1 5 SCREEN. IT MUST BE STARTED EVERY DAY BY THE OPERATOR.

---RESPONSIBILITY CATEGORY---

1 1 STATUS=CURRENT FUNCTION=DEVELOPED
PERSON=PROFIT PROGRAMMER

---NAMES CATEGORY---

1 1 MOCCLE NAME=PROFIT-PROG

---STORAGE CATEGORY---

1 1 TYPE=CORE IMAGE PLACE=P.F.
MECILP=DISK STORED SIZE= 1000 BYTES

---OPERATING CATEGORY---

1 1 REGION= 1

---RELATIONAL CATEGORY---

CATALOGUE NAME	PARTIAL	I/O
1 1 PROFIT-REC		
1 5 OFN-FLS		
1 10 RD-IAP		
1 15 FINISH		
1 20 COMPUTE-PROFIT		

Figure 11-13. Usage Report (Sheet 1 of 2)

USAGE OF PROFIT-PROG

CATALOGUE NAME REV LINE TYPE

INTER-ACT

SYSTEM

---DESCRIPTION CATEGORY---

1 1 THE INTERACTIVE PROG MUST BE MANUALLY STARTED

---RESPONSIBILITY CATEGORY---

1 1 PERSON=OPERATOR

---NAME CATEGORY---

1 1 SYSTEM NAME=START OPERATION

---RELATIONAL CATEGORY---

CATALOGUE NAME PARTIAL

1 1 PRCFIT-FRCG

REQUEST CONTROL REPORT

 REPORT/FILE REQUEST NUMBER 10

*** ENTRY TYPE CONTROL TOTALS

TYPE OF ENTRY	NUMBER READ	OF PRINTED IN \$REPORT	ENTRIES WRITTEN ON \$FILE	BYPASSED (DUPLICATES)
MODULE	1	1	0	0
SYSTEM	1	1	0	0
** REQUEST TOTALS	2	2	0	0

*** FILE RECORD COUNT CONTROL TOTALS

** 6 RECORDS WERE WRITTEN ON THE INDEX FILE
 ** 0 RECORDS WERE WRITTEN ON THE USER FILE (\$FILE)

***** END CONTROL REPORT FOR REQUEST 10 *****

Figure 11-13. Usage Report (Sheet 2 of 2)

DATA CATALOGUE 2
THE DATA CATALOGUE MANUAL

C A T A C A L O G U E 2

R E Q U E S T C O N T R O L R E P O R T

RUN TOTALS

*** REQUEST CONTROL TOTALS

*** 11 \$REPORT REQUESTS WERE PROCESSED
*** 0 \$FILE REQUESTS WERE PROCESSED
*** 11 REQUESTS WERE PROCESSED

*** FILE RECCRD CCUNT CONTROL TOTALS

*** 109 RECORDS WERE WRITTEN ON THE INDEX FILE
*** 0 RECORDS WERE WRITTEN ON THE USER FILE (\$FILE)

***** END \$REPORT/FILE REQUEST PROCESSING *****

Figure 11-14. Request Control Report, Run Totals

DATA CATALOGUE 2

DATA CATALOGUE 2 REPORT DATE- 06/05/79 PAGE 3
C A T A L O G U E I N D E X REVISION NUMBER- 0
C A T A L O G U E N A M E DATE OF LAST REVISION- 06/05/79

REQUEST NUMBER 2
\$REPORT CATALOGUE
TITLE = CLASS CATALOGUE REPORT 2

DATA CATALOGUE NAME	ENTRY TYPE	PAGE
BACK-ORDER	ELEMENT	21
CHECKSUM	ELEMENT	22
COMPUTE-PROFIT	MCCULE	3
CREATE	MCCULE	4
DESCRIPTION	ELEMENT	23
FINISH	MCCULE	6
FORM-A3	FORM	15
IN-STOCK-REPORT	REPORT	12
INTER-ACT	SYSTEM	2
INVENTORY	FILE	17
LEAD-TIME	ELEMENT	24
NO-PART	ELEMENT	25
ON-ORDER	ELEMENT	26
OPN-FLS	MCCULE	7
PART-ERR	MCCULE	8
PART-NO	ELEMENT	27
PART-REC	RECORD	19
PROFIT-PROG	MCCULE	9
PROFIT-REC	RECORD	20
PROFIT-SCREEN	REPORT	14
PT1	MCCULE	13
RD-INP	MCCULE	11
REORDER-PT	ELEMENT	28
STATEST	ELEMENT	29
UNIT-COST	ELEMENT	30
UNIT-PRICE	ELEMENT	31
UNIT-WEIGHT	ELEMENT	32

*** E N C O F I N D E X ***

Figure 11-15. Index Example

NOS Operating System

Job statement.
USER control statement.
CHARGE control statement.
ATTACH,DCRET/UN=LIBRARY.
ATTACH,MAST1,MAST2,MAST3/NA.
DEFINE,PNCHFIL.
FILE,MAST1,F0=DA,EFC=3.
FILE,MAST2,F0=DA,EFC=3.
FILE,MAST3,F0=WA,EFC=3.
DCRET.
CRMEP,LO.
REWIND,PNCHFIL.
COPYSBF,PNCHFIL,OUTPUT.
PURGE,PNCHFIL.
EXIT.
DMP.
DMP,377777.
PURGE,PNCHFIL.
7/8/9 card

NOS/BE Operating System

Job statement.
ATTACH,MAST1,I0=DC2,PW=DC2.
ATTACH,MAST2,I0=DC2,PW=DC2.
ATTACH,MAST3,I0=DC2,PW=DC2.
ATTACH,DCRET.
REQUEST,PNCHFIL,PF.
FILE,MAST1,FJ=DA,EFC=3.
FILE,MAST2,FJ=DA,EFC=3.
FILE,MAST3,FJ=WA,EFC=3.
DCRET.
CRMEP,LO.
REWIND,PNCHFIL.
COPYSBF,PNCHFIL,OUTPUT.
EXIT.
DMP.
DMP,377777.
PURGE,PNCHFIL.
7/8/9 card

\$GENERATE COBOL
OPTIONS LIST=YES
SELECT INVENTORY
6/7/8/9 card

Figure 11-16. Generate Request Example

DATA CATALOGUE 2
GENERATION REPORT

\$GENERATE COBOL
OPTIONS LIST=YES
SELECT INVENTORY

GENERATION REPORT

CARD IMAGE 1.....8...2...6...0...4...8.....0.....3.....0

000000*INVENTORY *DEFINITION CREATED FROM DC2
000010*DATE OF GENERATION: 06/05/79
000020*MASTER FILE REVISION NUMBER= 0
000030*DATE OF LAST REVISION= 06/05/79
000040*

000050 FD INVENTORY
000060 RECORD CONTAINS 80 CHARACTERS
000070 LABEL RECORDS ARE OMITTED
000080 BLOCK CONTAINS 1600 CHARACTERS
000090 DATA RECORDS ARE
000100 PART-REC
000110 PROFIT-REC.
000120 01 PART-REC.
000130 03 PART-NO PICTURE X(6).
000140 03 NO-PART PICTURE 9(4).
000150 03 BACK-ORDER PICTURE 9(6).
000160 03 ON-ORDER PICTURE 9(4).
000170 03 REORDER-PT PICTURE 9(4).
000180 03 LEAD-TIME PICTURE 999.
000190 03 LEAD-TIME PICTURE 9999V99.
000200 03 UNIT-COST PICTURE 9999V99.
000210 03 UNIT-PRICE PICTURE 9999V99.
000220 03 DESCRIPTION PICTURE X(17).
000230 03 CHECKSUM
000240 PICTURE 9(8) COMP-1
000250 SYNCHRONIZED.
000260 01 PROFIT-REC.
000270 03 PART-NO PICTURE X(6).
000280 03 UNIT-COST PICTURE 9999V99.
000290 03 UNIT-PRICE PICTURE 9999V99.

GENERATION REPORT

*** TOTAL GENERATION COMMANDS ATTEMPTED= 1

*** TOTAL COMMANDS ACCEPTED= 1

*** TOTAL COMMANDS REJECTED= 0

GENERATION REPORT

*** END DATA CATALOGUE RETRIEVAL REPORTS ***

Figure 11-17. Generation Report

NOS Operating System

Job statement.
USER control statement.
CHARGE control statement.
ATTACH,MAST3/NA.
ATTACH,DCCONVT,DCCONGN/LN=LIBRARY.
GET,CONVFIL.
FILE,WRKFILE,BT=C,RT=W,MNR=120,MRL=258.
FILE,TEMP1,BT=C,RT=W,MNR=120,MRL=258.
DCCONVT.
SORTMRG(O=VMSG/R)
DCCONGN.
REWIND,PNCHFIL.
COPYSBF,PNCHFIL,OUTPUT.
GRMEP,LO.
EXIT.
DMP.
DMP,377777.
GRMEP,LO.
7/8/9 card

NOS/BE Operating System

Job statement.
ATTACH,MAST3,ID=DC2,PT=DC2.
ATTACH,DCCONGN.
ATTACH,DCCONVT.
ATTACH,CONVFIL,ID=DC2.
FILE,WRKFILE,BT=C,RT=W,MNR=120,MRL=258.
FILE,TEMP1,BT=C,RT=W,MNR=120,MRL=258.
DCCONVT.
SORTMRG(O=VMSG/R)
DCCONGN.
REWIND,PNCHFIL.
COPYSBF,PNCHFIL,OUTPUT.
GRMEP,LO.
EXIT.
DMP.
DMP,377777.
GRMEP,LO.
7/8/9 card

```
$CONVERT DCCONL
MODULE CREATE INCLUDE NOTES DATNAME IS CREATE
SELECT FD
SELECT PART-REC AT LEVEL 01
7/8/9 card
```

```
SORT
FILE,INPUT=WRKFILE(R),OUTPUT=TEMP1(CR)
FIELD,SRTKEY(1,75,DISPLAY)
KEY,SRTKEY(A,C050L6)
END
6/7/8/9 card
```

Figure 11-18. Convert Request Example

DATA CATALOGUE 2
CONVERSION EDIT REPORT

CONVERSION CONTRL CARDS

\$CONVERT COEOL

MODLLE CREATE INCLLDE NOTES CATNAME IS CREATE
SELECT FD
SELECT PART-REC AT LEVEL 01

1 MODULES ACCEPTED FOR CONVERSION
0 MODULES REJECTED
1 MODULE READ

COBOL CONVERSION REPORT

PROGRAM ID	COEOL	CATANAME	SELECTED	IMAGE OF ITEM PROCESSED
CREATE		PART-REC		01 PART-REC.
				03 PART-NO PIC X(6).
				03 NO-PART PIC 9(4).
				03 BACK-ORDER PIC 9(6).
				03 CN-ORDER PIC 9(4).
				03 REORDER-PT PIC 9(4).
				03 LEAD-TIME PIC 999.
				03 UNIT-WEIGHT PIC 9999V99.
				03 UNIT-COST PIC 9999V99.
				03 UNIT-PRICE PIC 9999V99.
				03 DESCRIPTION PIC X(17).
				03 CHECKSUM PIC 9(8) USAGE COMP-2.

END COBOL CONVERSION REPORT

CONVERSION CONTROL REPORT

\$UPDATE

ADD ELE=BACK-ORDER

DES

1 THIS ENTRY WAS CREATED BY THE CONVERT FACILITY ON 16/05/79

NAM

1 DAT=BACK-ORDER

ATT

1 LEN=0006, FOR=N, PIC=9(6)

ADD ELE=CHECKSUM

DES

1 THIS ENTRY WAS CREATED BY THE CONVERT FACILITY ON 06/05/79

NAM

1 DAT=CHECKSUM

ATT

1 LEN=0008, FOR=F, PIC=9(8)

⋮

Figure 11-19. Conversion Reports (Sheet 1 of 2)

C O N V E R S I O N C O N T R O L R E P O R T

1 CAT=UNIT-WEIGHT
ATT
1 LEN=0006, FOR=N, PIC=9999V99

11 ELE ENTRIES HAVE BEEN GENERATED

C O N V E R S I O N C O N T R O L R E P O R T

ADD REC=PART-REC
CES
1 THIS ENTRY WAS CREATED BY THE CONVERT FACILITY ON 16/05/79
NAM
1 CAT=PART-REC
STR
0005 CAT=PART-NO
0010 CAT=NO-PART
0015 CAT=EACK-ORDER
0020 CAT=CN-ORDER
0025 CAT=RECRDER-PT
0030 CAT=LEAD-TIME
0035 CAT=LNIT-WEIGHT
0040 CAT=LNIT-COST
0045 CAT=LNIT-PRICE
0050 CAT=DESCRIPTION
0055 CAT=CHECKSUM

1 REC ENTRIES HAVE BEEN GENERATED

C O N V E R S I O N C O N T R O L R E P O R T

ADD MOD=CREATE
CES
1 THIS ENTRY WAS CREATED BY THE CONVERT FACILITY ON 16/15/79
NAM
1 NAM=CREATE
ATT
1 LAN=C

1 MOD ENTRIES HAVE BEEN GENERATED

C O N V E R S I O N C O N T R O L R E P O R T

*** END OF CONVERSION CONTROL REPORT ***

Figure 11-19. Conversion Reports (Sheet 2 of 2)

Control Data operating systems offer the following variations of a basic character set:

CDC 64-character set

CDC 63-character set

ASCII 64-character set

ASCII 63-character set

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

Depending on another installation option, the system assumes an input deck has been punched either in 026 or in 029 mode (regardless of the character set in use).

Under NOS/BE, the alternate mode can be specified by a 26 or 29 punched in columns 79 and 80 of the job

statement or any 7/8/9 card. The specified mode remains in effect throughout the job unless it is reset by specification of the alternate mode on a subsequent 7/8/9 card.

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

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

TABLE A-1. STANDARD CHARACTER SETS

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

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

TABLE A-2. CDC CHARACTER SET COLLATING SEQUENCE

Collating Sequence Decimal/Octal		CDC Graphic	Display Code	External BCD	Collating Sequence Decimal/Octal		CDC Graphic	Display Code	External BCD
00	00	blank	55	20	32	40	H	10	70
01	01	<	74	15	33	41	I	11	71
02	02	%	63 †	16 †	34	42	v	66	52
03	03	[61	17	35	43	J	12	41
04	04	→	65	35	36	44	K	13	42
05	05	≡	60	36	37	45	L	14	43
06	06	^	67	37	38	46	M	15	44
07	07	↑	70	55	39	47	N	16	45
08	10	↓	71	56	40	50	O	17	46
09	11	>	73	57	41	51	P	20	47
10	12	>	75	75	42	52	Q	21	50
11	13]	76	76	43	53	R	22	51
12	14	.	57	73	44	54]	62	32
13	15)	52	74	45	55	S	23	22
14	16	;	77	77	46	56	T	24	23
15	17	+	45	60	47	57	U	25	24
16	20	\$	53	53	48	60	V	26	25
17	21	*	47	54	49	61	W	27	26
18	22	-	46	40	50	62	X	30	27
19	23	/	50	21	51	63	Y	31	30
20	24	,	56	33	52	64	Z	32	31
21	25	(51	34	53	65	:	00 †	none†
22	26	=	54	13	54	66	0	33	12
23	27	≠	64	14	55	67	1	34	01
24	30	<	72	72	56	70	2	35	02
25	31	A	01	61	57	71	3	36	03
26	32	B	02	62	58	72	4	37	04
27	33	C	03	63	59	73	5	40	05
28	34	D	04	64	60	74	6	41	06
29	35	E	05	65	61	75	7	42	07
30	36	F	06	66	62	76	8	43	10
31	37	G	07	67	63	77	9	44	11

†In installations using the 63-graphic set, the % graphic does not exist. The : graphic is display code 63, External BCD code 16.

TABLE A-3. ASCII CHARACTER SET COLLATING SEQUENCE

Collating Sequence Decimal/Octal		ASCII Graphic Subset	Display Code	ASCII Code	Collating Sequence Decimal/Octal		ASCII Graphic Subset	Display Code	ASCII Code
00	00	blank	55	20	32	40	@	74	40
01	01	!	66	21	33	41	A	01	41
02	02	"	64	22	34	42	B	02	42
03	03	#	60	23	35	43	C	03	43
04	04	\$	53	24	36	44	D	04	44
05	05	%	63†	25	37	45	E	05	45
06	06	&	67	26	38	46	F	06	46
07	07	'	70	27	39	47	G	07	47
08	10	(51	28	40	50	H	10	48
09	11)	52	29	41	51	I	11	49
10	12	*	47	2A	42	52	J	12	4A
11	13	+	45	2B	43	53	K	13	4B
12	14	,	56	2C	44	54	L	14	4C
13	15	-	46	2D	45	55	M	15	4D
14	16	.	57	2E	46	56	N	16	4E
15	17	/	50	2F	47	57	O	17	4F
16	20	0	33	30	48	60	P	20	50
17	21	1	34	31	49	61	Q	21	51
18	22	2	35	32	50	62	R	22	52
19	23	3	36	33	51	63	S	23	53
20	24	4	37	34	52	64	T	24	54
21	25	5	40	35	53	65	U	25	55
22	26	6	41	36	54	66	V	26	56
23	27	7	42	37	55	67	W	27	57
24	30	8	43	38	56	70	X	30	58
25	31	9	44	39	57	71	Y	31	59
26	32	:	00†	3A	58	72	Z	32	5A
27	33	;	77	3B	59	73	[61	5B
28	34	<	72	3C	60	74	\	75	5C
29	35	=	54	3D	61	75]	62	5D
30	36	>	73	3E	62	76	^	76	5E
31	37	?	71	3F	63	77	_	65	5F

† In installations using a 63-graphic set, the % graphic does not exist. The : graphic is display code 63.

Diagnostics can be issued by any of the six Data Catalogue functions. A diagnostic is usually associated with an input transaction and is listed on the same report as the transaction.

The following types of diagnostics can be issued:

Fatal

The requested function cannot be initiated or it has been terminated prematurely. Fatal errors are usually caused by the absence of a function request, which is corrected by entering the proper function request; fatal errors can also occur if the Catalogue master files have been impaired. This type of error usually necessitates restoration of previously backed up files. A fatal error is always indicated by the ON condition for pseudo-sense switch 4 (SW4).

Serious

A particular transaction cannot be processed; for example, a nonexistent entry cannot be changed. The next transaction is read and normal processing continues. A serious error is always indicated by the ON condition for pseudo-sense switch 3 (SW3).

Warning

A particular field within a transaction cannot be processed. For example, a field contains an illegal code value. The field is bypassed and normal processing continues. Warning errors are always bypassed and normal processing continues. A warning error is always indicated by the ON condition for pseudo-sense switch 3 (SW3).

Information

The message is only informative. It does not indicate an error.

Data Catalogue diagnostics are output in the following message format:

function-number-severity *ERROR message text

The first part of the message identifies the function being requested, the unique error number for that function, and

the severity of the error. The five-character function identifiers are as follows:

DCUPD	Update function
DCQRY	Query function
DCRPT	Report function
DCGEN	Generate function
DCCVT	Convert function
DCUTL	Utility function

Error severity levels are indicated by the following codes:

F	Fatal
S	Serious
W	Warning
I	Information

The following example of the first part of an error message indicates serious error number 510 diagnosed by the Update function:

DCUPD-510-S

Tables B-1 through B-5 list the diagnostics issued by the functions as follows:

B-1	Update function
B-2	Query function
B-3	Report function
B-4	Generate function
B-5	Convert function

Diagnostics issued by the Utility function are contained in the data administrator's reference manual.

The tables show in tabular format the error number, severity level, and message text output by the functions. Two additional columns in the tables indicate the significance of the error and the action to be taken to correct the error.

TABLE B-1. UPDATE FUNCTION (DCUPD) DIAGNOSTICS

Error Number	Severity	Message	Significance	Action
010	I	CHANGE KEYWORD, OLD VALUE=xxxx	Documents a change to a field.	No action required.
020	I	DELETE KEYWORD, OLD VALUE=xxxx	Documents the deletion of a field.	No action required.
030	I	DELETE SUCCESSFULLY COMPLETED	Documents the deletion of an entry.	No action required.
040	I	DELETE OF REFERENCE TO THIS ENTRY IN catname	Documents deletion of an entry from the named higher level entry by a DELETE WHEREUSED statement.	No action required.
050	I	DELETE catname REF IN - catname	Documents deletion of an entry from the lower level entry by a DELETE USING statement.	No action required.
060	I	DELETE catname	Successful deletion of a lower level entry during execution of a DELETE USING or GLOBAL statement.	No action required.
200	W	keyword catname NOT FOUND	A Catalogue name defined in an ALIAS, STRUCTURE, or RELATIONAL category does not exist.	Ensure that the specified Catalogue name is correct or that the entry has been defined in the Catalogue.
205	W	keyword catname HIERARCHY ERROR	An attempt has been made to structure a higher level entry (e.g. system) into a lower level entry (e.g., module).	Ensure that the specified Catalogue name is correct or that the entry has been defined in the Catalogue.
210	W	keyword SAME NAME AS ENTRY CATALOGUE NAME	An attempt has been made to structure an entry into itself.	Ensure that the specified Catalogue name is correct.
215	W	NO VALUE ACCEPTED OR ALL DELETED	All fields in a category line were rejected.	Check the statement for incorrect field names or values.
220	W	keyword NO VALUE ON FILE	An attempt has been made to delete a nonexistent field.	Ensure that the specified field was correct.
225	W	keyword VALUE NOT NUMERIC	The specified field requires a numeric value.	Specify a numeric value for the field.
230	W	keyword SENSITIVE FIELD DELETED	A field defined by the data administrator as sensitive was deleted.	Reenter the field.
235	W	keyword VALUE IS AN INVALID CODE CODES ALLOWED-xxxx	The value entered for the specified field is not a valid code.	Enter one of the indicated valid codes for the field.
400	S	SYNTAX-NO ENTRY HEADER	The Update request did not include an ADD, CHG, or DEL statement.	Enter the request again with an ADD, CHG, or DEL statement.
405	S	UNKNOWN ENTRY TYPE	The ADD, CHG, or DEL statement did not specify a valid entity type.	Correct the ADD, CHG, or DEL statement.

TABLE B-1. UPDATE FUNCTION (DCUPD) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
410	S	SYNTAX-UNABLE TO LOCATE ENTRY TYPE	The ADD, CHG, or DEL statement did not specify an entity type.	Correct the ADD, CHG, or DEL statement.
415	S	CATALOGUE NAME LONGER THAN 32	The specified Catalogue name contains more than 32 characters.	Correct the Catalogue name.
420	S	ADD FOR EXISTING ENTRY	The ADD statement specifies a Catalogue name that already exists in the Catalogue.	Select a different Catalogue name.
425	S	CHG FOR NON-EXISTING ENTRY	The CHG statement specifies a Catalogue name that does not exist in the Catalogue.	Correct the Catalogue name specified in the statement.
430	S	DEL FOR NON-EXISTING ENTRY	An attempt was made to delete an entry that does not exist in the Catalogue.	Correct the Catalogue name specified in the statement.
435	S	SYNTAX-DEL TYPE ONLY ALLOWED FOR FUNC DEL	Extraneous information was discovered after the Catalogue name on an ADD or CHG statement.	Correct the ADD or CHG statement.
440	S	DELETE MUST BE WHEREUSED USING OR GLOBAL	In a DEL statement, Catalogue name is not followed by spaces, WHEREUSED, USING, or GLOBAL.	Correct the DEL statement.
445	S	CATALOGUE NAME IS RESERVED WORD	The specified Catalogue name is a reserved word.	Correct the Catalogue name.
500	S	SYNTAX-NO CATEGORY HEADER	The ADD or CHG statement is not followed by a Category Header statement.	Supply the Category Header statement for the ADD or CHG statement.
505	S	UNKNOWN CATEGORY	The Category Header statement does not specify a valid category name.	Supply the correct category name.
510	S	CATEGORY NOT VALID FOR ENTRY	The Category Header statement specifies a category that is not valid for the entity type.	Supply the correct category name.
515	S	SYNTAX-UNKNOWN KEYWORD IN CATEGORY HEADER	The Category Header statement contains a keyword that is not valid.	Correct the Category Header statement.
520	S	CATEGORY CANNOT BE DELETED	An attempt was made to delete a required category.	Correct the delete request.
525	S	CATEGORY DOES NOT EXIST IN ENTRY	The specified category is invalid for this entity type.	Enter a valid category name.
530	S	DELETE ONLY ALLOWED FOR FUNC CHG	Category or line deletions are not allowed for an ADD or DEL statement.	Correct the request.
600	S	NO LINE FOUND FOR LAST CATEGORY	A line deletion is not allowed when the category is empty.	Correct the CHG statement.
605	S	LINE NUMBER NOT NUMERIC	The specified line number is not a numeric value.	Correct the line number.
610	S	SYNTAX-LINE NUMBER NOT FOLLOWED BY SPACE	The line number is not followed by a space.	Insert a space after the line number.

TABLE B-1. UPDATE FUNCTION (DCUPD) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
615	S	LINE NUMBER OUT OF SEQUENCE	Within a category, line numbers must be entered in ascending order.	Correct the line number.
620	S	END OF FILE DURING LINE CONTINUATION	End-of-file was sensed on INPUT while searching for a continuation line.	Correct the continuation line.
625	S	SYNTAX-THRU MISSING ON LINE DELETE RANGE	When a range of line numbers is being deleted, the keyword THRU must be specified.	Correct the statement to include the keyword THRU.
630	S	THRU LINE NUMBER LOWER THAN LINE NUMBER	The second line number in a range specification must be greater than first line number.	Correct the line number.
635	S	SYNTAX-NO CLAUSE FOUND AFTER LINE NUMBER	The line was blank following the line number.	Enter the phrase for the line number.
640	S	LINE NUMBER DOES NOT EXIST IN CATEGORY	An attempt was made to change or delete a nonexistent line.	Correct the line number.
645	S	SYNTAX-ILLEGAL LINE CONTINUATION	A line continuation was not specified correctly or was specified for a line that cannot be continued.	Correct the continuation line.
650	S	keyword NOT A VALID KEYWORD	The indicated word is not a valid keyword for the statement.	Correct the statement.
655	S	keyword KEYWORD NOT VALID FOR CATEGORY	The indicated keyword is not valid for the category being processed.	Enter a correct keyword.
660	S	keyword VALUE IS TOO LONG	The value specified for the indicated keyword exceeds the maximum length.	Correct the value.
670	S	keyword UNABLE TO PROCESS KEYWORD	An internal program problem has been encountered.	Notify the system analyst.
675	S	keyword UNABLE TO PROCESS CATEGORY	An internal program problem has been encountered.	Notify the system analyst.
680	S	keyword DELETE ILLEGAL FOR FUNC ADD	Line or category deletions are not permitted while adding an entry.	Correct the statement.
685	S	keyword KEYWORD SEARCH EXTENDS PAST COL. 72	Transaction processing ends at column 72.	Correct the statement.
690	S	keyword VALUE SEARCH EXTENDS PAST COL. 72	Transaction processing ends at column 72.	Correct the statement.
700	S	NOT DELETED DUE TO REFERENCE IN - Catalogue name	The entry to be deleted was referenced in the named entries.	Delete references to the entry if it is to be deleted.
705	S	NOT DELETED DUE TO REFERENCE IN - Catalogue name TO Catalogue name	The entry to be deleted had components that were referenced in other entries.	Delete the references to the components if the entry is to be deleted.

TABLE B-1. UPDATE FUNCTION (DCUPD) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
800	S	MAST1-CTL READ MAST1-CTL REWRITE MAST1-CTL WRITE MAST1-DEL READ Catalogue name MAST1-DEL REWRITE Catalogue name	These messages indicate a problem with the Catalogue file MAST1.	Notify the data administrator.
850	S	MAST2-CTL READ MAST2-CTL WRITE MAST2-CTL ADD PTR MAST2-CTL DEL PTR MAST2-DEL DELETE Catalogue name MAST2-DEL DEL-PTR Catalogue name	These messages indicate a problem with the Catalogue file MAST2.	Notify the data administrator.
900	F	\$UPDATE TRANSACTION MISSING	The \$UPDATE statement is missing.	Enter the \$UPDATE statement.
910	F	OPTION TRANSACTION MISSING	The OPTIONS statement is missing.	Enter the OPTIONS statement.
915	F	REVISION NUMBER MUST BE 5 DIGITS	The revision number specified in the OPTIONS statement must consist of five digits.	Correct the revision number in the statement.
920	F	REVISION NUMBER NOT NUMERIC	A character other than a digit was specified for the revision number in the OPTIONS statement.	Correct the revision number in the statement.
925	F	INCORRECT REVISION NUMBER	The revision number specified in the OPTIONS statement is not a valid number.	Correct the revision number in the OPTIONS statement.
930	F	EDIT-ONLY VALUE MUST BE YES OR NO	The EDIT-ONLY phrase in the OPTIONS statement can specify only YES or NO.	Correct the EDIT-ONLY phrase in the statement.
935	F	UNKNOWN OPTION KEYWORD	A keyword specified in the OPTIONS statement is not known to Data Catalogue.	Correct the OPTIONS statement.
940	F	IMPACT-RPT VALUE MUST BE YES OR NO	The value for the IMPACT-RPT phrase must be either YES or NO.	Correct the IMPACT-RPT phrase.
950	F	MAST3-READ-CLIENT RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.
955	F	MAST3-READ-PW RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.
960	F	MAST3-READ-ENTRY RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.
965	F	MAST3-READ-CATG RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.
970	F	MAST3-READ-CLIENT FIELD RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.
975	F	MAST3-WRITE-CLIENT RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.

TABLE B-2. QUERY FUNCTION (DCQRY) DIAGNOSTICS

Error Number	Severity	Message	Significance	Action
005	I	COMMAND NOT FULLY EDITED	The error noted prevented the entire query command from being validated.	No action required.
010	I	LAST WORD PROCESSED SHOWN BELOW	The error noted prevented the entire query command from being validated. The last word of the command that was processed appears below this message.	No action required.
015	I	REQUESTED LINES NOT PRESENT FOR CATEGORY	A SHOW statement includes limit syntax specifying that the indicated lines are to be shown. Information exists for this entry in the designated category, but the requested lines are not present for this entry.	Correct the limit syntax.
020	I	REQUESTED CATEGORY NOT DEFINED	A SHOW statement includes limit syntax specifying that only the indicated category is to be shown for each entry. The requested category is not present for this entry.	No action required.
025	I	REQUESTED CATEGORIES NOT DEFINED	A SHOW statement includes limit syntax specifying that a category group is to be shown; e.g. ADMINISTRATION, CHARACTERISTICS, etc. None of these requested categories are present for this entry.	No action required.
400	S	FIRST STATEMENT NOT A FUNCTION	The first statement must be a \$QUERY statement.	Enter a \$QUERY statement.
405	S	NO COMMAND SPECIFIED	No COUNT, LIST, or SHOW statement was specified for the Query request. At least one of these statements must follow the \$QUERY statement.	Enter a COUNT, LIST, or SHOW statement.
410	S	STATEMENT SEQUENCE ERROR	The statements in the Query request are not in the proper order.	Arrange the statements in the proper order.
415	S	TITLE VALUE INVALID	The TITLE statement does not include a title, or the title is not entered enclosed in quotes.	Correct the TITLE statement.
420	S	TOO MANY CONTINUATION STATEMENTS	A statement cannot use more than two continuation lines. More than two have been specified for this statement.	Enter the statement using only two continuation lines.
425	S	CATALOGUE NAME NOT ON FILE	The statement specifies a Catalogue name, but the associated entry is not defined in the Catalogue.	Ensure that the correct Catalogue name has been specified.
430	S	INCOMPLETE LIMIT CLAUSE	The limit phrase does not include required words.	Correct the limit phrase of the SHOW statement.

TABLE B-2. QUERY FUNCTION (DCQRY) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
435	S	SHOW CATEGORY INVALID	A SHOW statement contains an invalid category name in the limit syntax.	Correct the limit syntax of the SHOW statement.
440	S	LINE NUMBER MUST BE NUMERIC	A SHOW statement contains an invalid line number in the limit phrase.	Correct the line number in the SHOW statement limit phrase.
445	S	ALIAS NOT VALID FOR NAMED ENTRY	An alias query can only be used for elements, groups, or records. The statement references an entity type that is not one of the above types.	Correct the statement to reference a valid entity type.
450	S	VERSION NOT VALID FOR NAMED ENTRY	A version query can only be used for procedural entities or a user entity. The statement identified an entity type that is not one of the above types.	Correct the statement to reference a valid entity type.
455	S	ALIAS/VERSION QUERY MUST USE-OF	The word OF is missing in an alias or version query.	Enter the word OF in the statement.
460	S	FIELD NAME INVALID	The statement includes a field name that is not valid. The statement includes a WITH clause that tests a group of entries (e.g., DATA, PROCEDURES, ENTRIES), but a field has been tested that is not in a common category (e.g., CONTROL, CLASSIFICATION, etc.).	Enter a field name for a common category.
465	S	INVALID WITH CLAUSE SYNTAX	A WITH clause does not contain a valid logical connector (e.g., EQ, NE).	Enter a valid logical connector.
470	S	ALPHA VALUE MISSING QUOTE	A WITH clause or range query includes a value that has a quotation mark on the left side but none on the right.	Enter the ending quotation mark.
475	S	FIELD VALUE MUST BE NUMERIC	A WITH clause tests a field with a numeric format, but the comparison value is not numeric.	Enter a numeric comparison value.
480	S	FIELD VALUE TOO LONG	A WITH clause contains a value that is longer than the field against which the value is to be compared.	Enter a value of the correct length.
485	S	FIRST WITH SEARCH CLAUSE INVALID	The previously indicated WITH clause syntax error was found on the first condition test.	Ensure that the first condition test is correct.
490	S	SECOND WITH SEARCH CLAUSE INVALID	The previously indicated WITH clause syntax error was found in the second condition test.	Ensure that the second test is correct.
495	S	THIRD WITH SECOND CLAUSE INVALID	The previously indicated WITH clause syntax error was found in the second condition test.	Ensure that the second condition test is correct.

TABLE B-2. QUERY FUNCTION (DCQRY) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
500	S	INVALID HAVING CLAUSE SYNTAX	The statement includes a HAVING clause that is syntactically incorrect or includes an invalid category name.	Correct the HAVING clause.
505	S	INVALID WHICH-USE ENTRY TYPE RANGE	The type of entity that is used is hierarchically higher; e.g., the first entity type in the statement is hierarchically lower than the second (for example, LIST ELEMENTS WHICH-USE SYSTEMS).	Enter the correct entity type in the WHICH-USE clause.
510	S	ENTRY TYPE CANNOT USE NAMED ENTRY	The WHICH-USE clause in the statement is invalid. The specified Catalogue entity is hierarchically higher than the entity type specified (for example, LIST GROUPS WHICH-USE REPORT).	Enter the correct entity type in the WHICH-USE clause.
515	S	ENTRY TYPE NOT USED-BY NAMED ENTRY	The USED-BY clause in the statement is invalid. The specified Catalogue entity is hierarchically lower than the entity type specified (for example, LIST RECORDS USED-BY ELEMENT-A).	Enter the correct entity type in the USED-BY clause.
520	S	INVALID USED-BY ENTRY TYPE RANGE	The entity type specified on the left side of the query is hierarchically higher than the entity type on the right side.	Enter the correct entity type in the USED-BY clause.
525	S	ENTRY TYPE RANGE INVALID	Valid entity types must be specified in both sides of the range.	Enter correct entity types.
535	S	NAME RANGE INVALID	The first word of the name range is an alphanumeric literal, which indicates that entities are to be retrieved between values. The word TO is missing.	Enter the word TO following the alphanumeric literal.
540	S	ENTRY RANGE MUST BE ASCENDING	The entity type specified on the right side of the query must be hierarchically greater than the type specified on the left.	Enter the correct entity type.
545	S	INVALID COMMAND SYNTAX	The statement wording is invalid.	Correct the statement.
550	S	EXCESS COMMAND WORDS	A valid and complete statement has been specified; however, an additional word follows.	Enter the command without the extra word.
555	S	INPUT IGNORED UNTIL NEXT FUNCTION	This message appears whenever statement sequence error occurs. No query statements are processed until the next \$QUERY statement is sensed.	Reenter the statements in the correct sequence.
560	S	NO INPUT SUBMITTED	A minimum of one \$QUERY statement and one command type of statement must be input. No input at all was sensed.	Enter a COUNT, LIST, or SHOW statement following the \$QUERY statement.

TABLE B-2. QUERY FUNCTION (DCQRY) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
800	F	MAST1 READ n	A retrieval error occurred while reading the Catalogue file MAST1. The Catalogue name of the entry that was being processed at the time the error was sensed is indicated in the message. Processing of the current statement was discontinued at the time the error was sensed. The next statement is processed.	Notify the data administrator.
805	F	MAST2 READ n	A retrieval error occurred while reading the Catalogue file MAST2. The Catalogue name of the entry that was being processed at the time the error was sensed is indicated in the message. Processing of the current statement was discontinued at the time the error was sensed. The next statement is processed.	Notify the data administrator.
950	F	MAST3 READ CLIENT RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.
955	F	MAST3 READ PW RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.
960	F	MAST3 READ CATG RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.
965	F	MAST3 READ FLD RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.
970	F	MAST3 READ HIT RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.
975	F	MAST3 READ ENT RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.

TABLE B-3. REPORT FUNCTION (DCRPT) DIAGNOSTICS

Error Number	Severity	Message	Significance	Action
015	I	REQUESTED LINES NOT PRESENT FOR CATEGORY	An OUTPUT statement includes a phrase specifying that the indicated category and lines are to be shown. Information exists in this entry for the designated category but the requested lines are not present for this entry.	No action required.
020	I	REQUESTED CATEGORY NOT DEFINED	An OUTPUT statement specifies that only the indicated category is to be shown for each entry. The requested category is not present for this entry.	No action required.

TABLE B-3. REPORT FUNCTION (DCRPT) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
025	I	REQUESTED CATEGORIES NOT DEFINED	An OUTPUT statement specifies that a category group is to be shown (for example, ADMINISTRATION, CHARACTERISTICS, etc.). None of the categories in the requested group is present for this entry.	No action required.
030	I	NO DATA SELECTED FOR REQUEST	The request has been completely processed but no entries qualified for selection.	No action required.
400	S	NO REQUEST INPUT	The retrieval module has been executed but no input statements were found.	Enter input statements for the request.
405	S	STATEMENT SEQUENCE ERROR	Statements in the Report request are not in the proper order.	Enter the statements in the correct sequence.
410	S	INPUT IGNORED UNTIL NEXT FUNCTION	This message appears whenever it is not possible to continue validation of a request (for example, after a sequence error). No statements are processed until the next function request statement is sensed.	Correct the statements in the Report request.
415	S	UNRECOGNIZABLE STATEMENT TYPE	A valid statement type has been found, but it is being used in the wrong context or it is out of order.	Enter the statement in the proper sequence.
425	S	NO REPORT/FILE TYPE SPECIFIED	A \$REPORT statement has been submitted, but the specific type of output report has not been specified.	Correct the \$REPORT statement to indicate the type of report being requested.
430	S	REPORT/FILE SYNTAX ERROR	The \$REPORT statement is not coded correctly.	Correct the \$REPORT statement.
435	S	request-type ILLEGAL REPORT/FILE	The type of output specified in the \$REPORT statement is not valid; request-type is the invalid word.	Correct the \$REPORT statement.
440	S	CATALOGUE NAME NOT ON FILE	The \$REPORT statement specifies a Catalogue name that is not defined in the Catalogue.	Correct the Catalogue name.
450	S	TITLE VALUE INVALID	The title specified for the request is too long, is not enclosed in quotes, or is spaces.	Correct the TITLE statement.
460	S	INVALID OPTION SYNTAX	The OPTIONS statement is not syntactically correct.	Correct the OPTIONS statement.
465	S	OPTION keyword HAS NO VALUE	The OPTIONS statement phrase indicated by keyword does not have any value specified.	Enter the correct value for the phrase.
470	S	keyword INVALID FOR REQUEST	The OPTIONS statement phrase indicated by keyword cannot be used for this type of request.	Correct the OPTIONS statement.

TABLE B-3. REPORT FUNCTION (DCRPT) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
475	S	keyword OPTION VALUE INVALID	The OPTIONS statement phrase indicated by keyword has a value that is not valid for the option.	Enter the correct value for the option.
485	S	TOO MANY OPTION CONTINUATIONS	The maximum number of continuation lines for the OPTIONS statement has been exceeded.	Reduce or eliminate the number of spaces in the statement.
500	S	SELECT STATEMENT IN ERROR WAS n	This message appears with other SELECT statement errors. It identifies the statement by number.	Correct the indicated statement.
505	S	INVALID SELECT SYNTAX	The indicated SELECT statement is syntactically incorrect.	Correct the SELECT statement.
510	S	TOO MANY SELECT STATEMENTS	More than nine SELECT statements have been specified for the request.	Correct the request to include no more than nine SELECT statements.
515	S	TOO MANY SELECT CONTINUATIONS	A SELECT statement has too many continuation lines.	Enter the SELECT statement with fewer continuation lines.
520	S	ALIAS/VERSION SELECT MUST USE-OF	The word OF is missing in an alias or version SELECT statement.	Enter the statement with the word OF.
525	S	ALIAS/VERSION INVALID FOR REQUEST	An alias or version SELECT statement has been included for a Hierarchy or Usage Report.	Enter the SELECT statement with a valid phrase.
530	S	ALIAS NOT VALID FOR NAMED ENTRY	An alias SELECT statement can only be used for elements, groups, and records. The statement references an entry that is not one of these types.	Enter the statement specifying a valid entry or use the VERSIONS OF phrase.
535	S	VERSION NOT VALID FOR NAMED ENTRY	A version SELECT statement is only valid for entries hierarchically greater than a record type entity.	Enter the statement specifying a valid entry or use the ALIASES OF phrase.
540	S	NAME RANGE INVALID	The first word of the SELECT statement is an alphanumeric literal, which indicates that entities to be retrieved are between values. The word TO is missing.	Enter the statement specifying the word TO.
545	S	NAME RANGE MUST BE ASCENDING	The name or value specified on the right side of the SELECT statement must be alphabetically greater than the name or value specified on the left side.	Enter the statement specifying valid names or values.
550	S	ENTRY TYPE RANGE INVALID	Valid entity types must be specified on both sides of the SELECT statement range.	Enter the statement with both entities of the same type.
555	S	ENTRY RANGE MUST BE ASCENDING	The entity type specified on the right side of the SELECT statement must be hierarchically greater than the entity type specified on the left.	Enter the statement specifying valid entity types.

TABLE B-3. REPORT FUNCTION (DCRPT) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
560	S	INVALID HIERARCHY ENTRY RANGE	The range of entities specified by the SELECT statement is not within the requested hierarchy (for example, an attempt to select files from the hierarchy of a record).	Enter the statement specifying valid entity types.
565	S	INVALID USAGE ENTRY RANGE	The range of entities specified by the SELECT statement is hierarchically invalid (for example, an attempt to select records that use a file).	Enter the statement specifying valid entity types.
570	S	ENTRY TYPE CANNOT USE NAMED ENTRY	The entity identified in the \$REPORT statement for a Usage Report request is hierarchically greater than the entity type indicated in the SELECT statement.	Correct the \$REPORT or SELECT statement.
575	S	ENTRY TYPE NOT USED BY NAMED ENTRY	The entity identified in the \$REPORT statement of a Hierarchy Report request is hierarchically lower than the entity type indicated in the SELECT statement.	Correct the \$REPORT or SELECT statement.
580	S	ALPHA VALUE MISSING QUOTE	A quotation mark was specified on the left side of a value but not on the right side.	Enter a quotation mark on the right side of the value.
585	S	FIELD NAME INVALID	The SELECT statement includes a field name that is not valid. The statement includes a WITH clause that tests a group of entities (for example, GDATA, GPROCEDURES, ENTRIES), but a field has been tested that is not in a common category (for example, CONTROL, CLASSIFICATION, etc.).	Correct the SELECT statement.
590	S	FIELD VALUE TOO LONG	A WITH clause contains a value that is longer than the field against which the value is to be compared.	Enter a correct value in the WITH clause.
595	S	FIELD VALUE MUST BE NUMERIC	The WITH clause tests a field with a numeric format, but the comparison value is not numeric.	Enter a numeric value in the WITH clause.
600	S	INVALID WITH CLAUSE SYNTAX	The WITH clause does not contain a valid logical connector (for example, EG or NE).	Enter a valid logical connector in the WITH clause.
605	S	INVALID HAVING CLAUSE SYNTAX	The SELECT statement includes a HAVING clause that is syntactically incorrect or includes an invalid category name.	Correct the HAVING clause in the statement.
610	S	FIRST WITH SEARCH CLAUSE INVALID	The previously indicated WITH clause syntax error was found in the first condition test.	Ensure that the WITH clause is correct.
615	S	SECOND WITH SEARCH CLAUSE INVALID	The previously indicated WITH clause syntax error was found in the second condition test.	Ensure that the WITH clause is correct.

TABLE B-3. REPORT FUNCTION (DCRPT) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
620	S	THIRD WITH SEARCH CLAUSE INVALID	The previously indicated WITH clause syntax error was found in the second condition test.	Ensure that the WITH clause is correct.
625	S	SELECT REFERS TO MISSING OUTPUT	The SELECT statement contains a USE-OUTPUT phrase, but the OUTPUT statement has not been specified.	Enter the OUTPUT statement or correct the SELECT statement.
630	S	INVALID USE-OUTPUT CLAUSE	The SELECT statement contains a USE-OUTPUT phrase without an output statement number.	Enter the output statement number.
635	S	EXCESS STATEMENT WORDS	The SELECT statement is complete, but one or more additional words appear to the right of the statement.	Enter the statement without the excess words.
700	S	OUTPUT STATEMENT IN ERROR WAS n	This message appears to further document other messages. The number of the OUTPUT statement that contains the error is indicated in this message.	Correct the indicated OUTPUT statement.
705	S	REFERS TO SELECT STATEMENT n	An invalid OUTPUT statement has been processed. The category is not valid for the type of entity specified in the indicated SELECT statement.	Correct the OUTPUT statement.
710	S	INVALID OUTPUT STATEMENT	The OUTPUT statement does not contain the literal CATEGORY or CAT, contains an unreferenced output statement number, or is missing a required output statement number.	Correct the OUTPUT statement.
715	S	TOO MANY OUTPUT STATEMENTS	The maximum of nine OUTPUT statements has been exceeded.	Delete the excess OUTPUT statements.
720	S	OUTPUT STATEMENTS MUST BE NUMBERED	A USE-OUTPUT phrase was specified in a SELECT statement, but none of the OUTPUT statements are numbered.	Enter an output statement number for the applicable OUTPUT statement.
725	S	OUTPUT STATEMENTS ILLEGALLY NUMBERED	No SELECT statement in the request specifies the USE-OUTPUT phrase, but one or more OUTPUT statements are numbered.	Enter a SELECT statement with the USE-OUTPUT phrase, or eliminate the output statement number.
730	S	OUTPUT STATEMENT SEQUENCE ERROR	An OUTPUT statement is out of place or is not numbered properly.	Enter the statements in the proper sequence, or correct the output statement numbers.
735	S	INVALID LINE LIMITS	The OUTPUT statement contains a syntax error, or category line numbers are not properly specified.	Correct the OUTPUT statement.
740	S	OUTPUT CATEGORY CLAUSE ILLEGAL	The OUTPUT statement contains a syntax error or an invalid category.	Correct the OUTPUT statement.

TABLE B-3. REPORT FUNCTION (DCRPT) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
745	S	INCOMPLETE LINE LIMITS	The OUTPUT statement indicates that specific lines are to be retrieved for the category, but the specification is incomplete.	Correct the OUTPUT statement.
750	S	OUTPUT CATEGORY INVALID	The category defined in the OUTPUT statement is not a valid category or is being referenced illegally.	Correct the OUTPUT statement.
755	S	OUTPUT LINE NUMBER MUST BE NUMERIC	The OUTPUT statement indicates that specific lines are to be retrieved for a category, but the specified line number is not numeric.	Enter a numeric line number.
760	S	DUPLICATE OUTPUT CATEGORIES	Categories must be unique for a given output set (for example, a category cannot be output more than once for a given SELECT statement).	Correct the OUTPUT statement.
800	S	MAST1 READ n	A retrieval error occurred while reading the Catalogue file MAST1. The Catalogue name of the entry that was being processed at the time the error was sensed is indicated in the error message.	Notify the data administrator.
805	S	MAST2 READ n	A retrieval error occurred while reading the Catalogue file MAST2. The Catalogue name of the entry that was being processed at the time the error was sensed is indicated in the error message.	Notify the data administrator.
850	S	INDEX FILE SEQUENCE ERROR	The index output of the DCRPT module must be sorted prior to input to the DCIDX module.	Perform the sort operation before calling the DCIDX module into execution.
860	S	NO INDEX FILE DATA	An end-of-file occurred on the first attempt to read the file; the DCRPT module and a sort operation must be executed before executing the DCIDX module. The sorted index file is written on the work file TEMP15.	Execute the DCRPT module and the sort operation before executing the DCIDX module.
870	S	NO REQUEST FILE DATA	An end-of-file occurred on the first attempt to read the work file TEMP11.	Notify the data administrator.
880	S	EXTRACT FILE SEQUENCE ERROR	TEMP12, which is created by DCSEL, must be sorted before the DCRPT module is executed.	Perform the sort operation.
890	S	NO EXTRACT FILE DATA	An end-of-file occurred on the first attempt by the DCSEL module to read the work file TEMP12. A sort operation must be executed before the DCRPT module is executed.	Perform the sort operation.

TABLE B-3. REPORT FUNCTION (DCRPT) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
950	F	MAST3 READ CLIENT RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.
960	F	MAST3 READ CATG RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.
965	F	MAST3 READ FLD RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.
970	F	MAST3 READ ENT RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.

TABLE B-4. GENERATE FUNCTION (DCGEN) DIAGNOSTICS

Error Number	Severity	Message	Significance	Action
400	S	GENERATE LANGUAGE MISSING	The \$GENERATE statement does not specify COBOL or DBDL. The entire request is bypassed.	Correct the \$GENERATE statement.
405	S	n IS AN INVALID LANGUAGE NAME	The \$GENERATE statement specifies a name that is not valid. The entire request is bypassed.	Correct the \$GENERATE statement.
410	S	STATEMENT SEQUENCE ERROR	The statements of the Generate request are not in order, or the request includes an invalid statement type.	Correct the statements in the request.
415	S	INVALID OPTION SYNTAX	The syntax of the OPTIONS statement is not correct. The entire request is rejected.	Correct the OPTIONS statement.
420	S	OPTION n HAS ILLEGAL VALUE	The OPTIONS statement option indicated in the message specifies a value that is not correct for that option. The entire request is bypassed.	Correct the option specified in the statement.
425	S	n MUST BE NUMERIC	The OPTIONS statement option indicated in the message requires that a numeric value be specified. The entire request is bypassed.	Correct the option specified in the statement.
430	S	n INVALID FOR LANGUAGE USED	The OPTIONS statement option indicated in the message is not valid for the language type specified in the \$GENERATE statement.	Correct the option specified in the statement.
435	S	LAST OPTION FOLLOWED BY COMMA	The last OPTIONS statement phrase is followed by a comma. A comma should not be specified after this phrase. The entire request is bypassed.	Delete the comma following the last phrase in the OPTIONS statement.

TABLE B-4. GENERATE FUNCTION (DCGEN) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
440	S	INVALID OPTIONS CONTINUATION	When an OPTIONS statement is continued, the last phrase in each line except the last line must be followed by a comma; column 1 of each continuation line must be blank. The entire request is bypassed.	Correct the OPTIONS statement.
445	S	TOO MANY CONTINUATION STATEMENTS	There can be only two continuation lines for an OPTIONS statement; this limit has been exceeded. The entire request is bypassed.	Correct the OPTIONS statement so that only two continuation lines are needed.
450	S	n NUMERIC VALUE TOO LARGE	The OPTIONS statement option indicated in the message requires a numeric value; the specified value is too large. The entire request is bypassed.	Enter a numeric value of the correct length for the option.
455	S	NO COMMAND SPECIFIED	At least one SELECT statement must be specified for a Generate request; no SELECT statements were read.	Enter at least one SELECT statement.
460	S	NO RECORD OR FILE NAMED FOR SELECT	The SELECT statement required phrase is not specified. The name of a file or record must be specified. Processing of the SELECT statement is bypassed.	Correct the SELECT statement.
465	S	catname NOT ON FILE	The SELECT statement requests the generation of a data definition for an item that is not contained in the Catalogue. Processing of the SELECT statement is bypassed.	Correct the SELECT statement.
470	S	RECORD OR FILE MUST BE SELECTED	The SELECT statement requests the generation of a data definition for an entry that is stored in the Catalogue, but the entry is not a file or record. Processing of the SELECT statement is bypassed.	Correct the SELECT statement.
480	S	INVALID DATA BASE SELECT COMMAND	The syntax of a SELECT statement for a DBDL Generate request is not valid. Processing of the SELECT statement is bypassed.	Correct the SELECT statement.
485	S	SELECT MUST BE FOR DATA BASE ENTITY	A SELECT statement for a DBDL Generate request must specify a data base entity. Processing of the SELECT statement is bypassed.	Correct the SELECT statement.
490	S	v INVALID FOR n	The value indicated by v is invalid for the DBDL option indicated by n. The entire request is bypassed.	Correct the Generate request.
495	S	INPUT IGNORED UNTIL NEXT FUNCTION	Because of the previously noted errors, the remaining statements of the Generate request have been bypassed. Processing resumes with the next function request statement.	Correct the indicated errors.

TABLE B-4. GENERATE FUNCTION (DCGEN) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
500	S	catname HAS NO DATA	During generation of the requested data definition statements, the indicated entry has been retrieved from the Catalogue, but it does not contain any information other than a Catalogue name. No statement is generated for the entry. Generation processing continues.	Enter the required information in the Catalogue.
525	S	catname IS NOT A GROUP OR ELEMENT ENTITY	During generation of a COBOL data definition, a record or group structure line encountered contains the name of a Catalogue entry, but the indicated entry is neither a group nor an element. No statement is generated for the entry. Generation processing continues.	Correct the entity definition in the Catalogue.
530	S	catname IS NOT A RECORD ENTITY	During generation of a COBOL data definition, a file structure line encountered contains the name of a Catalogue entry, but the indicated entry is not a record. No statement is generated for the entry. Generation processing continues.	Correct the entity definition in the Catalogue.
540	I	DEFAULT VALUES USED TO CREATE PICTURE BELOW	The elementary entry for which the PICTURE clause was generated does not define format and/or length attributes. The PICTURE clause has been formatted based on default values.	No action required.
545	S	GROUP-LENGTH CANNOT BE CALCULATED	The DS instruction to save the storage area for a group cannot be calculated due to faulty group definition.	Review and correct the entity definitions.
550	S	GROUP LEVEL EXCEEDS LIMIT	The COBOL generation limit of ten group levels has been exceeded. Generation processing continues with the next valid group or record.	Review the definition of these groups for errors and/or redefine the entities.
560	I	DEFAULT VALUES USED TO CREATE DEFINITION BELOW	The length field for an element was set to spaces or zero; the default of 1 was used.	No action required.
580	I	DEFINITION NOT GENERATED FOR catname	A definition could not be fully generated because required data has not been entered in the Catalogue.	No action required.
585	I	DEFAULT FORMAT USED TO CREATE DEFINITION BELOW	The format field was spaces; the default of C was used to allow generation.	No action required.
590	S	LENGTH INVALID FOR FORMAT	An element format field cannot have the length specified in the element entity length field; normally, the field is too large.	No action required.

TABLE B-4. GENERATE FUNCTION (DCGEN) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
595	S	INITIAL VALUE INVALID FOR FORMAT	An element format type cannot contain the initial value specified in the element entity initial field.	No action required.
600	S	catname HAS NO STRUCTURE	During generation of data definitions, an entry encountered does not have any structure lines. The error message indicates the Catalogue name of this entry. Generation processing continues.	Review Catalogue definitions.

TABLE B-5. CONVERT FUNCTION (DCCVT) DIAGNOSTICS

Error Number	Severity	Message	Significance	Action
400	S	NO COMMANDS SPECIFIED	No request statements encountered; the run is terminated.	Enter the correct request statements.
405	S	ILLEGAL BLANK CARD	Blank cards are bypassed until the next nonblank card is encountered.	No action required.
410	S	CARD OUT OF SEQUENCE	Request statements are not in the following order: COBOL: \$CONVERT MODULE SELECT (multiple) RENAME (optional) CREATE (optional) TOTAL: \$CONVERT SELECT (multiple) RENAME (multiple, optional) CREATE (optional)	Enter the statements in the correct order.
415	S	INCOMPLETE \$CONVERT CARD	The \$CONVERT statement contains a syntax error; the run is terminated.	Correct the \$CONVERT statement.
420	S	language-name IS AN INVALID LANGUAGE NAME	The \$CONVERT statement does not specify COBOL or DBDL; the run is terminated.	Correct the \$CONVERT statement.
425	S	NO PROGRAM NAME FOR MODULE CARD	A program name is not specified in the MODULE statement. Statements are bypassed until a MODULE statement for COBOL conversion or a SELECT statement for TOTAL conversion is encountered.	Correct the MODULE statement.
430	S	SELECT CARD INCOMPLETE/ INCORRECT	The SELECT statement contains a syntax error. Statements are bypassed until a MODULE statement for COBOL conversion or a SELECT statement for TOTAL conversion is encountered.	Correct the SELECT statement.

TABLE B-5. CONVERT FUNCTION (DCCVT) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
435	S	PROGRAM TABLE EXCEEDS LIMIT	The limit of 100 programs specified for conversion has been exceeded. The first 100 modules are processed.	Enter a request to process the excess programs.
440	S	SELECT TABLE EXCEEDS LIMIT	The limit of 200 selections has been exceeded. The first 200 selections are processed.	Enter a request to process the excess selections.
445	S	MODULE CARD INCOMPLETE/ INCORRECT	The MODULE statement contains a syntax error. Statements are bypassed until a MODULE statement for COBOL conversion or a SELECT statement for TOTAL conversion is encountered.	Correct the MODULE statement.
450	S	NO DATANAME FOR SELECT CARD	The SELECT statement has no data name specified. Statements are bypassed until a MODULE statement for COBOL conversion or a SELECT statement for TOTAL conversion is encountered.	Correct the SELECT statement.
455	S	RENAMES TABLE EXCEEDS LIMIT	The limit of 200 RENAME statements has been exceeded. The first 200 RENAME statements are processed.	Enter a request to process the excess RENAMES statements.
460	S	ILLEGAL LEVEL NUMBER	The level number specified is not FD or numeric. Statements are bypassed until a MODULE statement for COBOL conversion or a SELECT statement for TOTAL conversion is encountered.	Enter FD or a numeric value for the level number.
465	S	INPUT IGNORED UNTIL NEXT FUNCTION	Statements are bypassed until a MODULE statement for COBOL conversion or a SELECT statement for TOTAL conversion is encountered.	Correct the bypassed statements.
470	S	TOO MANY CONTINUATION STATEMENTS	The limit of three continuation statements has been exceeded. Statements are bypassed until a MODULE statement for COBOL conversion or a SELECT statement for TOTAL conversion is encountered.	Correct the continuation statements.
475	S	RENAME CARD INCOMPLETE/ INCORRECT	The RENAME statement contains a syntax error. Statements are bypassed until a MODULE statement for COBOL conversion or a SELECT statement for TOTAL conversion is encountered.	Correct the RENAME statement.
480	S	ILLEGAL LINE NUMBER	The line number indicated by the BEGIN NUMBER AT phrase is zero or nonnumeric.	Correct the BEGIN NUMBER AT phrase.
485	S	CREATE CARD INCOMPLETE/ INCORRECT	The CREATE statement contains a syntax error. Statements are bypassed until a MODULE statement for COBOL conversion or a SELECT statement for TOTAL conversion is encountered.	Correct the CREATE statement.

TABLE B-5. CONVERT FUNCTION (DCCVT) DIAGNOSTICS (Contd)

Error Number	Severity	Message	Significance	Action
490	S	TRUNCATION OCCURRED IN PICTURE/VALUE ABOVE	A picture or value is truncated to 25 bytes, the length of the field in the Catalogue.	No action required.
495	S	program-name NOT ON FILE	The program name for the requested module was not found in scanning the source program.	Ensure that the request statements are correct.
500	W	ENTRY WITH CATALOGUE NAME OF FILLER BYPASSED	An entry assigned the Catalogue name FILLER was bypassed.	No action required.
550	F	MAST3 READ CLIENT RECORD	A problem has been encountered with the control file (MAST3) in the Catalogue.	Notify the data administrator.

- Alias -**
An element, group, or record entry that is established as a variation of another entry in the Catalogue. A category of information in an element entity used to document alternate names and descriptions for an element.
- Catalogue -**
A computerized dictionary of data, procedures, and users; the master files that contain all the information related to the entity definitions. The Catalogue consists of a data file, a relational file, and a control file. It is a central repository where information processing documentation can be maintained, accessed, retrieved, and distributed at computer speeds.
- Catalogue Name -**
The unique key that identifies and relates all of the information concerning an entry in the Catalogue. Each entry must be assigned a unique Catalogue name.
- Category -**
A class of information that is used to define an entity to Data Catalogue. Several categories exist and are used in various combinations to define different entity types. A category contains one or more fields of information.
- Convert Function -**
The Data Catalogue procedure used to create entity definitions from existing COBOL programs and TOTAL DBDLs. Input transactions are generated by the Convert function for processing by the Update function.
- Data Base Entity -**
A collection of information that describes the characteristics of a TOTAL data base. A data base is a logical aggregate of data; it is the collection of datasets that participate in a defined relationship.
- Data Dictionary -**
Synonymous with Catalogue.
- Data Entities -**
A class of entity types for defining data structures. Element, group, record, file, dataset, and data base entity types are data entities.
- Dataset Entity -**
The entity type for describing the characteristics of a TOTAL dataset. A dataset is a file of records in the TOTAL data base environment. It is a part of the data base and can be related to other datasets within the data base.
- Element Entity -**
The entity type for defining data that cannot be subdivided. An element is the basic building block of data structures. It is the lowest level data entity in the hierarchy. A TOTAL data item is described to Data Catalogue as an element entity.
- Entity -**
A Data Catalogue component consisting of various categories of information. Entities are grouped into classes according to the type of information used to describe them. This information can include definitions, values, technical attributes, control information, usage data, structural definitions, and relationships.
- Entry -**
A discrete occurrence of an entity type in the Catalogue.
- External Resource Entity -**
The entity type used to define an information processing resource that is not accessible by computer. Relevant resources could be flowcharts or procedural instructions.
- Field -**
A unit of information in an entity definition. One or more fields are provided for each category applicable to an entity type.
- File Entity -**
The entity type for describing the characteristics of a file, which is a collection of one or more records that can be accessed and stored physically. A single file can contain several different types of records and many different records of each type.
- Form Entity -**
The entity type containing information related to a form used in the information processing cycle. A form is a document containing printed captions, data, and blank spaces that need to be filled with data. It can be a source for input data, or it can be partly the output of one Data Catalogue function and partly the source of another function.
- Function -**
A process that performs a specific type of operation for creating, maintaining, and accessing the Catalogue. The six Data Catalogue functions provided are Update, Query, Report, Generate, Convert, and Utility.
- Generate Function -**
The Data Catalogue procedure used to create source statement data definitions from information in the Catalogue. COBOL, PL/I, and TOTAL DBDL statements can be generated.
- Group Entity -**
The entity type for describing a logical collection of elements and subgroups. A group contains two or more subcomponents, which can be elements and/or other groups. A TOTAL element is described to Data Catalogue as a group entity.
- Hierarchy -**
The rank or order of entity types recognized by Data Catalogue. An element entity is the lowest level in the hierarchy and a user entity is the highest level in the hierarchy.

Manual Task Entity -

The entity type for describing an activity that does not involve any computer processing. Checking control totals is a manual task.

Module Entity -

The entity type for describing a computer process that performs one or more discrete tasks. A module can be free-standing or part of a program.

Procedure Entities -

A class of entity types for describing the processes that perform a task and the resources required to complete the task. Procedure entities are one step up the hierarchy from data entities. Module, program, system, form, report, external resource, and manual task entity types are procedure entities.

Program Entity -

Synonymous with module entity. A program can contain modules, call modules, or consist of one module by itself.

Query Function -

The Data Catalogue procedure used to interrogate the Catalogue. Entries can be counted, listed by Catalogue name, or listed in detail.

Record Entity -

The entity type for describing the characteristics of a record. A record is a logically associated collection of related elements and/or groups.

Report Entity -

The entity type for describing a formatted presentation of information that can be printed or displayed. A report is the end product of the information processing cycle and can be prepared manually or by computer.

Report Function -

The Data Catalogue procedure used to produce various reports based on information in the Catalogue. The reports are used for documentation and for analysis.

Report Name -

The name by which a category field is identified on Data Catalogue reports.

Request -

A series of statements that identify the function to be performed and specify the requirements for the desired operation.

Screen Entity -

Synonymous with Report Entity.

System Entity -

The entity type for describing an information processing system. A system is a self-contained collection of one or more computer programs and ancillary manual tasks that perform a given function completely. A system contains processes that in turn access resources.

System Name -

The name specified by the user when entering or accessing a category field.

Update Function -

The Data Catalogue procedure used to maintain entries in the Catalogue. Entries can be added, changed, or deleted through this function.

User Entity -

The entity type for defining responsibility of a management unit for a given activity or resource. A management unit can be assigned responsibility for a system, a program, a manual task, a file, or an external resource.

Utility Function -

The Data Catalogue procedure used to initialize and support the Catalogue. This is a data administrator function.

Version -

A procedure or user entry that is established as a variation of another entry in the Catalogue.

SUMMARY OF ENTITY DEFINITIONS

D

An entity definition consists of various fields of information in a number of categories. The categories and the fields within categories that can be used to define an entity depend on the entity type. A summary of applicable fields is presented for each entity type in the following tables:

- D-1 Element entity
- D-2 Group entity
- D-3 Record entity
- D-4 File entity
- D-5 Data base entity
- D-6 Dataset entity
- D-7 Form entity
- D-8 Report entity
- D-9 External resource entity
- D-10 Task entity
- D-11 Module (program) entity
- D-12 System entity
- D-13 User entity

The columns in the tables are defined as follows:

Category

The name of the category to which the fields belong.

System Name

The field name as it is specified by the user in Data Catalogue statements.

Report Name

The field name as it is printed by Data Catalogue in reports.

Maximum Length

The maximum number of characters in the field.

Edit

The type of edit performed by Data Catalogue:

- No editing of the field
- * Must be one of the characters listed
- Length Must not exceed the field length
- Numeric Must be a numeric value

Input Methods

The applicable methods for defining the field: user transactions, Convert function, Generate function.

Notes

Meaning of one-character edited code or other explanation.

TABLE D-1. ELEMENT ENTITY

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
CONTROL	EALIASOF	ALIAS OF	32	Length	X	X	X	Exists Proposed Obsolete User assigned
	VERSION	VERSION OF	32	Length	X	X	X	
	ESTATUS	STATUS	1	* E P 0 0-9	X			
CLASSIFICATION				-				User defined
DESCRIPTION				-				Free form
OTHER	88LEVEL	88 LEVEL	67	-		X	X	

TABLE D-1. ELEMENT ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
ORIGIN	DEPT	DEPARTMENT	32	Length	X			Field is generated Field not generated User assigned Daily Weekly Monthly Quarterly Semiannually Annually Requested User assigned
	PROGRAM	PROGRAM	32	Length	X			
	FORM	FORM	32	Length	X			
	GENCODE	GENERATED	1	* Y N 0-9	X			
	CYCLE	CYCLE		1 D W M Q S A R 0-9	*	X		
	TIMELIFE	EXPECTED LIFE	3	Numeric	X			
	UNITLIFE	Depends on code seleted	1	* D W M Q S A R 0-9	X		Days Weeks Months Quarters Semiannual periods Annual periods User assigned	
VALUES	VALUE	VALUE	52	-	X			
	NOTE	NOTE	15	-	X			
NAMES	DATANAME	DATA NAME	32	Length	X	X	X	
	IDEN	IDENTIFIER	31	Length	X	X		
	SYMBOL	SYMBOL	8	Length	X	X	X	
	DBMSNAME	DBMS NAME	8	Length	X	X	X	
ATTRIBUTES	LENGTH	LENGTH	4	Numeric	X	X	X	
	FORMAT	FORMAT	1	* B C F H N O P 0-9	X	X	X	Binary Character Floating point Hexadecimal Numeric Octal Packed User assigned
	JUST	JUSTIFIED	1	* L R 0-9	X	X	X	Left Right User assigned
	SYNC	SYNCHRONIZED	1	* Y N 0-9	X	X	X	Synchronized Not synchronized User assigned

TABLE D-1. ELEMENT ENTITY

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes	
					User	Convert	Generate		
	VOLATILE	VOLATILITY	1	* D S 0-9	X			Dynamic Static User assigned	
	NATURE	NATURE	8	-	X				
	MEASURE	UNIT OF MEASURE	8	-	X				
	PICTURE	PICTURE	25	-	X	X	X		
	IVALUE	INIT VALUE	25	-	X	X	X		
ALIAS	ADATANAM	DATA NAME	32	Length	X	X	X	Binary Character Floating point Hexadecimal Numeric Octal Packed User assigned	
	AIDEN	IDENTIFIER	31	Length	X	X			
	ASYMBOL	SYMBOL	8	Length	X	X	X		
	ALENGTH	LENGTH	4	Numeric	X	X	X		
	AFORMAT	FORMAT	1	* B C F H N O P 0-9	X	X	X		
	AJUST	JUSTIFIED	1	* L R 0-9	X	X	X		Left Right User assigned
	ALIASYNC	SYNCHRONIZED	1	* Y N	X	X	X		Synchronized Not synchronized
	APICTURE	PICTURE	25	-	X	X	X		
	AINIT	INIT VALUE	25	-	X	X	X		

TABLE D-2. GROUP ENTITY

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
CONTROL	EALIAS	ALIAS OF	32	Length	X	X	X	Exists Proposed Obsolete User assigned
	VERSION	VERSION OF	32	Length	X	X	X	
	ESTATUS	STATUS	1	* E P O 0-9	X			
CLASSIFICATION				-				User defined

TABLE D-2. GROUP ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes	
					User	Convert	Generate		
DESCRIPTION				-				Free form	
OTHER	88LEVEL	88 LEVEL	67	-		X	X		
NAMES	DATANAME	DATA NAME	32	Length	X	X	X		
	IDEN	IDENTIFIER	31	Length	X	X			
	SYMBOL	SYMBOL	8	Length	X	X	X		
	DBMSNAME	DBMS NAME	8	Length	X	X	X		
ATTRIBUTES	LENGTH	LENGTH	4	Numeric	X	X	X	Binary Character Floating point Hexadecimal Numeric Octal Packed User assigned	
	FORMAT	FORMAT	1	* B C F H N O P 0-9	X	X	X		
	PICTURE	PICTURE	25	-	X	X	X		
	JUST	JUSTIFIED	1	* L R 0-9	X	X	X		Left Right User assigned
	SYNC	SYNCHRONIZED	1	* Y N 0-9	X	X	X		Synchronized Not synchronized User assigned
STRUCTURE	CATNAME	CATALOGUE NAME	32	Length	X	X	X	OCCURS FROM OCCURS TO Duplicate keys OK Unique sequential Multiple sequential Not a key field User assigned	
	ALIAS	ALIAS	4	Numeric	X	X	X		
	FILLER	FILLER	4	Numeric	X	X	X		
	REDEFINE	REDEFINES	32	Length	X	X	X		
	FROM	FROM	3	Numeric	X	X	X		
	TO	TO	3	Numeric	X	X	X		
	INDEX	INDEXED BY	32	Length	X	X	X		
	DEPEND	DEPENDING ON	32	Length	X	X	X		
	KEY	KEY	1	* D U M N 0-9	X				
	INC	INCLUSION	1	* Y N 0-9	X	X			Included/generation Not included User assigned

TABLE D-3. RECORD ENTITY

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
CONTROL	EALIASOF	ALIAS OF	32	Length	X	X	X	Exists Proposed Obsolete User assigned
	VERSION	VERSION OF	32	Length	X	X	X	
	ESTATUS	STATUS	1	* E P O 0-9	X			
CLASSIFICATION				-				User defined
DESCRIPTION				-				Free form
OTHER	88LEVEL	88 LEVEL	67	-		X	X	
NAMES	DATANAME	DATA NAME	32	Length	X	X	X	
	IDEN	IDENTIFIER	31	Length	X	X	X	
	SYMBOL	SYMBOL	8	Length	X	X	X	
ATTRIBUTES	MAXLEN	MAXIMUM LENGTH	4	Numeric	X	X	X	Based Controlled Static Automatic User assigned
	MINLEN	MINIMUM LENGTH	4	Numeric	X	X	X	
	STORAGE	STORAGE	1	* B C S A 0-9	X	X		
STRUCTURE (Conventional record)	CATNAME	CATALOGUE NAME	32	Length	X	X	X	OCCURS FROM OCCURS TO Duplicate keys OK Unique sequential Multiple sequential Not a key field User assigned Yes, include No, exclude User assigned
	ALIAS	ALIAS	4	Numeric	X	X	X	
	FILLER	FILLER	4	Numeric	X	X	X	
	REDEFINE	REDEFINES	32	Length	X	X	X	
	FROM	FROM	3	Numeric	X	X	X	
	TO	TO	3	Numeric	X	X	X	
	INDEX	INDEXED BY	32	Length	X	X	X	
	DEPEND	DEPENDING ON	32	Length	X	X	X	
	KEY	KEY	1	* D U M N 0-9	X			
INC	INCLUSION	1	* Y N 0-9	X	X			
STRUCTURE (TOTAL record)	TKEY	KEY	1	* Y N 0-9	X	X	X	Yes, a TOTAL key No, not a TOTAL key User assigned

TABLE D-3. RECORD ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
	LINKPATH	LINKAGE PATH	6	Length	X	X	X	Enter mmmccc
	VARLEN	LENGTH/VARIABLE	5	Numeric	X	X		
	RCCODE	REC CODE	2	Length	X	X	X	
	COMPNAME	COMPONENT	32	Length	X	X	X	

TABLE D-4. FILE ENTITY

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
CONTROL	EALIASOF	ALIAS OF	32	Length	X	X	X	Exists Proposed Obsolete User assigned
	VERSION	VERSION OF	32	Length	X	X	X	
	ESTATUS	STATUS	1	* E P O 0-9	X			
CLASSIFICATION				-				User defined
DESCRIPTION				-				Free form
OTHER	88LEVEL	88 LEVEL	67	-		X	X	
RESPONSIBILITY	STATUS	STATUS	1	* C P 0-9	X			Current Past User assigned Developed Changed Tested User assigned
	FUNCTION	FUNCTION	1	* D C T 0-9	X			
	DEPT	DEPARTMENT	32	-	X			
	PERSON	PERSON	32	-	X			
	PHONE	PHONE	10	-	X			
	TITLE	TITLE	15	-	X			
	MAIL	MAIL	5	-	X			
DATE	DATE	6	-	X				
NAMES	FDNAME	FDNAME	32	Length	X	X	X	
	DSNAME	DATA SET NAME	44	Length	X			
	MEMBER	MEMBER NAME	8	Length	X			
	DDNAME	DDNAME	8	Length	X	X	X	

TABLE D-4. FILE ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
ATTRIBUTES	RECSIZE	RECSIZE	5	Numeric	X	X	X	Direct access Direct access unmovable Indexed sequential Indexed seq. unmovable Partitioned Partitioned unmovable Physical sequential Physical seq. unmovable Standard labels ANS labels ANS and ANS user labels No labels Standard & user labels Nonstandard labels Bypass label processing 200 bpi 7 track 556 bpi 7 track 800 bpi 7 and 9 track 1600 bpi 9 track 6250 bpi 9 track User assigned Direct addressing Indirect addressing User assigned
	BLKSIZE	BLKSIZE	5	Numeric	X	X	X	
	RFM	RECFM	6	-	X			
	DSORG	DSORG	3	*Length DA DAU IS ISU PO POU PS PSU	X			
	LABEL	LABEL	3	*Length SL AL AUL NL SUL NSL BLP	X			
	DENSITY	DENSITY	1	* 0 1 2 3 4 5-9	X			
	ADDRESS	ADDRESS MODE	1	* D I 0-9	X			
	RMODNAME	RMOD NAME	32	Length	X			
	KEYLEN	KEYLEN	3	Numeric	X			
	RKP	RKP	3	Numeric	X			
	CYLOFL	CYLOFL	2	Numeric	X			
	OPTCD	OPTCD	3	Length	X			
LIMCT	LIMCT	2	Numeric	X				
UNIT	PUNIT	UNIT	7	-	X			
	PVOLSER	VOLUME SERIAL	6	-	X			
	PSPACE	SPACE	3	Numeric	X			
	PTYPE	Depends on code selected	1	* C T B R D 0-9	X		Cylinders Tracks Blocks Records Directory blocks User assigned	

TABLE D-4. FILE ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
	IUNIT	IDX UNIT	7	-	X			Cylinders Tracks Blocks Records Directory blocks User assigned
	IVOLSER	IDX VOLUME SERIAL	6	-	X			
	ISPACE	IDX SPACE	3	Numeric	X			
	ITYPE	Depends on code selected	1	* C T B R D 0-9	X			
	OUNIT	OVFL UNIT	7	-	X			
	OVOLSER	OVFL VOLUME SERIAL	6	-	X			
	OSPACE	OVFL SPACE	3	Numeric	X			
	OTYPE	Depends on code selected	1	* C T B R D 0-9	X		Cylinders Tracks Blocks Records Directory blocks User assigned	
STRUCTURE	CATNAME	CATALOGUE NAME	32	Length	X	X	X	
	RECTYPE	TYPE	8	-	X			
	FREQ	FREQUENCY	7	Numeric	X			

TABLE D-5. DATA BASE ENTITY

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
CONTROL	EALIASOF	ALIAS OF	32	Length	X	X	X	Exists Proposed Obsolete User assigned
	VERSION	VERSION OF	32	Length	X	X	X	
	ESTATUS	STATUS	1	* E P O 0-9	X			
CLASSIFICATION				-				User defined
DESCRIPTION				-				Free form
OTHER	88LEVEL	88 LEVEL	67	-		X	X	
RESPONSIBILITY	STATUS	STATUS	1	* C P 0-9	X			Current Past User assigned

TABLE D-5. DATA BASE ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
	FUNCTION	FUNCTION	1	* C D T 0-9	X			Changed Developed Tested User assigned
	DEPT	DEPARTMENT	32	Length	X			
	PERSON	PERSON	32	Length	X			
	PHONE	PHONE	10	Length	X			
	TITLE	TITLE	15	Length	X			
	MAIL	MAIL	5	Length	X			
	DATE	DATE	6	Length	X			
NAMES	DBSNAME	DB NAME	8	Length	X	X	X	
	NAME	NAME	32	Length	X			
IOAREA	TRACK	TRACK HOLD	1	* Y N 0-9	X	X	X	Yes No User assigned
	IOAREA	IOAREA NAME	4	Length	X	X	X	
	OCCURS	OCCURS	2	Numeric	X	X	X	
STRUCTURE	DSNAME	DATASET NAME	32	Length	X			Master Variable User assigned
	TYPE	TYPE	1	* M V 0-9	X	X	X	
	EXCLUDE	EXCLUDE	1	* Y N 0-9	X	X		

TABLE D-6. DATASET ENTITY

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
CONTROL	EALIASOF	ALIAS OF	32	Length	X	X	X	Exists Proposed Obsolete User assigned
	VERSION	VERSION OF	32	Length	X	X	X	
	ESTATUS	STATUS	1	* E P O 0-9	X			
CLASSIFICATION				-				User defined

TABLE D-6. DATASET ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes	
					User	Convert	Generate		
DESCRIPTION				-				Free form	
OTHER	88LEVEL	88 LEVEL	67	-		X	X		
RESPONSIBILITY	STATUS	STATUS	1	* C P 0-9	X			Current Past User assigned	
	FUNCTION	FUNCTION	1	* C D T 0-9	X			Changed Developed Tested User assigned	
	DEPT	DEPARTMENT	32	Length	X				
	PERSON	PERSON	32	Length	X				
	PHONE	PHONE	10	Length	X				
	TITLE	TITLE	15	Length	X				
	MAIL	MAIL	5	Length	X				
	DATE	DATE	6	Length	X				
NAMES	TDSNAME	TOTAL DSNAME	4	Length	X	X	X		
	DSNAME	DS NAME	44	Length	X				
	DDNAME	DD NAME	8	Length	X				
	PRIME	PRIME	8	Length	X				
	MEMBER	MEMBER	8	Length	X				
ATTRIBUTES	RECSIZE	RECSIZE	5	Numeric	X			Direct access Direct access unmovable Indexed sequential Indexed seq. unmovable Partitioned Partitioned unmovable Physical sequential Physical seq. unmovable	
	BLKSIZE	BLKSIZE	5	Numeric	X				
	RFM	RECFM	6	Length	X				
	DSORG	DSORG	3	* DA DAU IS ISU PO POU PS PSU	X				
	LABEL	LABEL	3	* X SL AL AUL NL SUL NSL BLP					
									Standard labels ANS labels ANS & ANS user labels No labels Standard & user labels Nonstandard labels Bypass label processing

TABLE D-6. DATASET ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
	RMODNAME	RMOD NAME	32	Length	X			
	KEYLEN	KEYLEN	3	Numeric	X			
	RKP	RKP	3	Numeric	X			
	CYLOFL	CYLOFL	2	Numeric	X			
	OPTCD	OPTCD	3	Length	X			
	LIMCT	LIMCT	2	Numeric	X			
UNIT	PUNIT	UNIT	7	Length	X			
	PVOLSER	VOLUME SERIAL	6	Length	X			
	PSPACE	SPACE	3	Numeric	X			
	PTYPE	Depends on code selected	1	* C T B R D 0-9	X			Cylinders Tracks Blocks Records Directory blocks User assigned
	IUNIT	IDX UNIT	7	Length	X			
	IVOLSER	IDX VOLUME SERIAL	6	Length	X			
	ISPACE	IDX SPACE	3	Numeric	X			
	ITYPE	Depends on code selected	1	* C T B R D 0-9	X			Cylinders Tracks Blocks Records Directory blocks User assigned
	OUNIT	OVFL UNIT	7	Length	X			
	OVOLSER	OVFL VOLUME SERIAL	6	Length	X			
	OSPACE	OVFL SPACE	3	Numeric	X			
	OTYPE	Depends on code selected	1	* C T B R D 0-9	X			Cylinders Tracks Blocks Records Directory blocks User assigned
ENVIRONMENT	DEVICE	DEVICE	7	Length	X	X	X	
	TRECORDS	TOTAL RECORDS	7	Numeric	X	X	X	
	TTRACKS	TOTAL TRACKS	5	Numeric	X	X	X	
	TLENGTH	RECD LENGTH	5	Numeric	X	X	X	
	BLOCKS	BLOCKS/TRACK	5	Numeric	X	X	X	

TABLE D-6. DATASET ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
	TEXTENTS	# OF EXTENTS	2	Numeric	X	X	X	Yes No User assigned
	LOADPCT	LOAD % LIMIT	2	Numeric	X	X	X	
	OLDFILE	OLD FILE	1	* Y N 0-9	X	X	X	
	RELSTART	REL START	5	Numeric	X	X	X	
	VTCTRKS	VTOC TRACKS	5	Numeric	X	X	X	
STRUCTURE	CATNAME	CATALOGUE NAME	32	Length	X	X	X	
	RECTYPE	TYPE	8	Length	X			
	FREQ	FREQUENCY	7	Numeric	X			

TABLE D-7. FORM ENTITY

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
CONTROL	EALIASOF	ALIAS OF	32	Length	X	X	X	Exists Proposed Obsolete User assigned
	VERSION	VERSION OF	32	Length	X	X	X	
	ESTATUS	STATUS	1	* E P 0 0-9	X			
CLASSIFICATION				-				User defined
DESCRIPTION				-				Free form
OTHER	88LEVEL	88 LEVEL	67	-		X	X	
RESPONSIBILITY	STATUS	STATUS	1	* C P 0-9	X			Current Past User assigned
	FUNCTION	FUNCTION	1	* C D T 0-9	X			Changed Developed Tested User assigned
	DEPT	DEPARTMENT	32	Length	X			
	PERSON	PERSON	32	Length	X			
	PHONE	PHONE	10	-	X			
	TITLE	TITLE	15	-	X			

TABLE D-7. FORM ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
	MAIL	MAIL	5	-	X			
	DATE	DATE	6	-	X			
NAMES	NAME	FORM NAME	32	Length	X			
	NUMBER	FORM NUMBER	10	-	X			
ATTRIBUTES	PARTS	PARTS	2	Numeric	X			
	COLOR	COLOR	10	-	X			
SUPPLIER	SNAME	SUPPLIER NAME	32	Length	X			
	SUPNO	SUPPLIER NUMBER	10	-	X			
	QREORDER	REORDER POINT	5	Numeric	X			
	UREORDER	Depends on code selected	1	* B C H R T 0-9	X			Box Carton Hundred Ream Thousand User assigned
	QONHAND	ONHAND QTY	5	Numeric	X			
	UONHAND	Depends on code	1	* B C H R T 0-9	X			Box Carton Hundred Ream Thousand User assigned
	QEQQ	EQQ	5	Numeric	X			
	UEQQ	Depends on code selected	1	* B C H R T 0-9	X			Box Carton Hundred Ream Thousand User assigned
	QMINORD	MINIMUM ORDER QTY	5	Numeric	X			
	UMINORD	Depends on code selected	1	* B C H R T 0-9	X			Box Carton Hundred Ream Thousand User assigned
	STIME	LEAD TIME	3	Numeric	X			
	SUNIT	Depends on code selected	1	* D W M Q S A 0-9	X			Days Weeks Months Quarters Semiannual periods Annual periods User assigned

TABLE D-7. FORM ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
FLOW	FLOWCODE	FLOW	1	* O P U 0-9	X			Originated here Processed Used here User assigned
	FDEPT	DEPARTMENT	32	Length	X			
	FPERSON	PERSON	32	Length	X			
	FPHONE	PHONE	10	-	X			
	FPARTS	PARTS	2	Numeric	X			
	FSUBDAY	DAY	10	-	X			
RELATIONAL	CATNAME	CATALOGUE NAME	32	Length	X			Partial use Full use User assigned
	PARTIAL	PARTIAL	1	* Y N 0-9	X			

TABLE D-8. REPORT ENTITY

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
CONTROL	EALIASOF	ALIAS OF	32	Length	X	X	X	Exists Proposed Obsolete User assigned
	VERSION	VERSION OF	32	Length	X	X	X	
	ESTATUS	STATUS	1	* E P O 0-9	X			
CLASSIFICATION				-				User defined
DESCRIPTION				-				Free form
OTHER	88LEVEL	88 LEVEL	67	-		X	X	Current Past User assigned Changed Developed Tested User assigned
RESPONSIBILITY	STATUS	STATUS	1	* C P 0-9	X			
	FUNCTION	FUNCTION	1	* C D T 0-9	X			
	DEPT	DEPARTMENT	32	Length	X			
	PERSON	PERSON	32	Length	X			
	PHONE	PHONE	10	-	X			

TABLE D-8. REPORT ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
	TITLE	TITLE	15	-	X			
	MAIL	MAIL	5	-	X			
	DATE	DATE	6	-	X			
NAMES	NAME	REPORT NAME	32	Length	X			
	NUMBER	REPORT NUMBER	10	-	X			
	FORMNAME	FORM NAME	32	Length	X			
ATTRIBUTES	MEDIUM	MEDIUM	1	* D M P S T 0-9	X			Display Microfilm Printed Spooled Typed User assigned
	PARTS	PARTS	2	Numeric	X			
FLOW	FDEPT	DEPARTMENT	32	Length	X			} B Burst } D Decollated } M Microfiche } F Fastened } R Reduced } 0-9 User assigned Days Weeks Months Quarters Semiannual periods Annual periods Indefinitely User assigned
	FPERSON	PERSON	32	Length	X			
	FPHONE	PHONE	10	-	X			
	FCOPIES	COPIES	2	Numeric	X			
	F1MODE	MODE: 1	1	*	X			
	F2MODE	MODE: 2	1	*	X			
	F3MODE	MODE: 3	1	*	X			
	F4MODE	MODE: 4	1	*	X			
	F5MODE	MODE: 5	1	*	X			
	FDISTDAY	DAY	10	-	X			
	FRETAIN	RETENTION	3	Numeric	X			
FURETAIN	Depends on code selected	1	* D W M Q S A I 0-9	X				
RELATIONAL	CATNAME	CATALOGUE NAME	32	Length	X			Partial use Full use User assigned
	PARTIAL	PARTIAL	1	* Y N 0-9	X			
	LTYPE	TYPE	8	-	X			
	PICTURE	PICTURE	25	-	X			
	START	START	3	Numeric	X			
	END	END	3	Numeric	X			

TABLE D-9. EXTERNAL RESOURCE ENTITY

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
CONTROL	EALIASOF	ALIAS OF	32	Length	X	X	X	Exists Proposed Obsolete User assigned
	VERSION	VERSION OF	32	Length	X	X	X	
	ESTATUS	STATUS	1	* E P O 0-9	X			
CLASSIFICATION				-				User defined
DESCRIPTION				-				Free form
OTHER	88LEVEL	88 LEVEL	67	-		X	X	
RESPONSIBILITY	STATUS	STATUS	1	* C P 0-9	X			Current Past User assigned
	FUNCTION	FUNCTION	1	* C D T 0-9	X			Changed Developed Tested User assigned
	DEPT	DEPARTMENT	32	Length	X			
	PERSON	PERSON	32	Length	X			
	PHONE	PHONE	10	-	X			
	TITLE	TITLE	15	-	X			
	MAIL	MAIL	5	-	X			
	DATE	DATE	6	-	X			
NAMES	NAME	RESOURCE NAME	32	Length	X			
	ID	RESOURCE ID	10	-	X			
LOCATION	TYPRES	RESOURCE TYPE	1	* D F O L P S T 0-9	X			Documentation Flowchart Operating instructions Listing Procedural instructions Status report Test data User assigned
	LOCRES	LOCATION	66	-	X			
RELATIONAL	CATNAME	CATALOGUE NAME	32	Length	X			
	PARTIAL	PARTIAL	1	* Y N 0-9	X			Partial user Full use User assigned

TABLE D-10. MODULE (PROGRAM) ENTITY

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
CONTROL	EALIASOF	ALIAS OF	32	Length	X	X	X	Exists Proposed Obsolete User assigned
	VERSION	VERSION OF	32	Length	X	X	X	
	ESTATUS	STATUS	1	* E P O 0-9	X			
CLASSIFICATION				-				User defined
DESCRIPTION				-				Free form
OTHER	88LEVEL	88 LEVEL	67	-		X	X	
RESPONSIBILITY	STATUS	STATUS	1	* C P 0-9	X			Current Past User assigned
	FUNCTION	FUNCTION	1	* C D T 0-9	X			Changed Developed Tested User assigned
	DEPT	DEPARTMENT	32	Length	X			
	PERSON	PERSON	32	Length	X			
	PHONE	PHONE	10	-	X			
	TITLE	TITLE	15	-	X			
	MAIL	MAIL	5	-	X			
	DATE	DATE	6	-	X			
NAMES	NAME	MODULE NAME	32	Length	X			
	MODID	MODULE ID	10	Length	X			
ATTRIBUTES	LANGPRG	LANGUAGE	1	* A C F M P R 0-9	X			Assembler COBOL FORTRAN MARK IV PL/I RPG User assigned
	TYPEPRG	TYPE	1	* B T 0-9	X			Batch Teleprocessed User assigned
	OVLYPRG	OVERLAID	1	* Y N 0-9	X			Overlaid Not overlaid User assigned
	SEGMENTD	SEGMENTED	1	* Y N 0-9	X			Segmented Not segmented User assigned

TABLE D-10. MODULE (PROGRAM) ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
	CALLTYPE	CALL TYPE	1	* E R 0-9	X			Reentrant code Recursive code User assigned
STORAGE	STYPE	TYPE	1	* L O P R S C 0-9	X			Load module Object module Partitioned dataset Relocatable Source Core image User assigned
	PLACE	PLACE	20	-	X			
	MEDIUM	MEDIUM	20	-	X			
	SIZE	STORED SIZE	7	Numeric	X			
	UNITSIZE	Depends on code selected	1	* B C K P R T 0-9	X			Bytes Cylinders K bytes Pages Records Tracks User assigned
OPERATING	STEPNAME	STEP NAME	44	Length	X			
	REGION	REGION	4	Numeric	X			
	TELAPSED	ELAPSED TIME	3	Numeric	X			
	UELAPSED	Depends on code selected	1	* H M S 0-9	X			Hours Minutes Seconds User assigned
	TIMECPU	CPU TIME	3	Numeric	X			
	UNITCPU	Depends on code selected	1	* H M S 0-9	X			Hours Minutes Seconds User assigned
	NAMEPROC	PROC NAME	8	Length	X			
	SEQPROC	PROC SEQUENCE	2	Numeric	X			
	LINES	LINES PRINTED	7	Numeric	X			
PUNCH	CARDS PUNCHED	7	Numeric	X				

TABLE D-10. MODULE (PROGRAM) ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
SCHEDULING	NBEGIN	NORMAL START	4	Numeric	X			Daily Weekly Monthly Quarterly Semiannually Annually Request User assigned
	NEND	NORMAL STOP	4	Numeric	X			
	NDAY	NORMAL DAY	10	-	X			
	DBEGIN	DEADLINE START	4	Numeric	X			
	DEND	DEADLINE STOP	4	Numeric	X			
	DDAY	DEADLINE DAY	10	-	X			
	SCYCLE	CYCLE	1	* D W M Q S A R 0-9	X			
	PRECPRG	PRECEDING PROGRAM	32	Length	X			
PASSING	PASSING	PASSING	32	Length	X			
RELATIONAL	CATNAME	CATALOGUE NAME	32	Length	X			Partial use Full use User assigned Input only Output only Both input and output User assigned All accesses Creates Deletes Reads Updates User assigned Daily Weekly Monthly Quarterly Semiannually Annually Request User assigned
	PARTIAL	PARTIAL	1	* Y N 0-9	X			
	IOCODE	I/O	1	* I O B 0-9	X			
	ACCESS	ACCESS	1	* A C D R U 0-9	X			
	CYCLEPRG	CYCLE	1	* D W M Q S A R 0-9	X			
	FREQPRG	FREQ	5	Numeric	X			
	DISP	DISP	20	Length	X			

TABLE D-11. TASK ENTITY

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
CONTROL	EALIASOF	ALIAS OF	32	Length	X	X	X	Exists Proposed Obsolete User assigned
	VERSION	VERSION OF	32	Length	X	X	X	
	ESTATUS	STATUS	1	* E P O 0-9	X			
CLASSIFICATION				-				User defined
DESCRIPTION				-				Free form
OTHER	88LEVEL	88 LEVEL	67	-		X	X	
RESPONSIBILITY	STATUS	STATUS	1	* C P 0-9	X			Current Past User assigned
	FUNCTION	FUNCTION	1	* C D T 0-9	X			Changed Developed Tested User assigned
	DEPT	DEPARTMENT	32	Length	X			
	PERSON	PERSON	32	Length	X			
	PHONE	PHONE	10	-	X			
	TITLE	TITLE	15	-	X			
	MAIL	MAIL	5	-	X			
	DATE	DATE	6	-	X			
NAMES	NAME	TASK NAME	32	Length	X			
	ID	TASK ID	10	-	X			
SCHEDULING	NBEGIN	NORMAL START	4	Numeric	X			Daily Weekly Monthly Quarterly Semiannually Annually Request User assigned
	NEND	NORMAL STOP	4	Numeric	X			
	NDAY	NORMAL DAY	10	-	X			
	DBEGIN	DEADLINE START	4	Numeric	X			
	DEND	DEADLINE STOP	4	Numeric	X			
	DDAY	DEADLINE DAY	10	-	X			
	SCYCLE	CYCLE	1	* D W M Q S A R 0-9	X			
	PRECSTEP	PRECEDING STEP	32	Length	X			

TABLE D-11. TASK ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
RELATIONAL	CATNAME	CATALOGUE NAME	32	Length	X			Partial use Full use User assigned Input only Output only Both input and output User assigned Daily Weekly Monthly Quarterly Semiannually Annually Request User assigned
	PARTIAL	PARTIAL	1	* Y N 0-9	X			
	IOCODE	I/O	1	* I O B 0-9	X			
	CYCLE	CYCLE	1	* D W M Q S A R 0-9	X			

TABLE D-12. SYSTEM ENTITY

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
CONTROL	EALIASOF	ALIAS OF	32	Length	X	X	X	Exists Proposed Obsolete User assigned
	VERSION	VERSION OF	32	Length	X	X	X	
	ESTATUS	STATUS	1	* E P O 0-9	X			
CLASSIFICATION				-				User defined
DESCRIPTION				-				Free form
OTHER	88LEVEL	88 LEVEL	67	-		X	X	
RESPONSIBILITY	STATUS	STATUS	1	* C P 0-9	X			Current Past User assigned
	FUNCTION	FUNCTION	1	* C D T 0-9	X			Changed Developed Tested User assigned
	DEPT	DEPARTMENT	32	Length	X			
	PERSON	PERSON	32	Length	X			
	PHONE	PHONE	10	-	X			
	TITLE	TITLE	15	-	X			

TABLE D-12. SYSTEM ENTITY (Contd)

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
	MAIL	MAIL	5	-	X			
	DATE	DATE	6	-	X			
NAMES	NAME	SYSTEM NAME	32	Length	X			
	ID	SYSTEM ID	10	-	X			
RELATIONAL	CATNAME	CATALOGUE NAME	32	Length	X			Partial use Full use User assigned
	PARTIAL	PARTIAL	1	* Y N 0-9	X			
	JID	JOB NAME	8	Length	X			
	JNAME	JOB TITLE	32	Length	X			

TABLE D-13. USER ENTITY

Category	System Name	Report Name	Maximum Length	Edit	Input Methods			Notes
					User	Convert	Generate	
CONTROL	EALIASOF	ALIAS OF	32	Length	X	X	X	Exists Proposed Obsolete User assigned
	VERSION	VERSION OF	32	Length	X	X	X	
	ESTATUS	STATUS	1	* E P O 0-9	X			
CLASSIFICATION				-				User defined
DESCRIPTION				-				Free form
OTHER	88LEVEL	88 LEVEL	67	-		X	X	
PERSONNEL	SECTION	SECTION	32	Length	X			Changed Developed Tested User assigned
	FUNCTION	FUNCTION	1	* C D T 0-9	X			
	PERSON	PERSON	32	Length	X			
	TITLE	TITLE	15	-	X			
	PHONE	PHONE	10	-	X			
	MAIL	MAIL	5	-	X			
RELATIONAL	CATNAME	CATALOGUE NAME	32	Length	X			Partial use Full use User assigned
	PARTIAL	PARTIAL	1	* Y N 0-9	X			
	RPERSON	PERSON	32	Length	X			

REQUEST FORMS

E

Sample forms that can be used for function requests are included in this appendix. These forms can be reproduced

or be used as a guide to design a master form. The use of request forms is discussed in section 2.

SUBMITTED BY

ORGANIZATION

DATE

SEQUENCE

1 FUNCTION STATEMENT 72

73 80

\$UPDATE

OPTIONS STATEMENT

OPTIONS

ENTRY STATEMENT

CATEGORY STATEMENT

LINE STATEMENTS

1 72

73 80

CONVERT REQUEST

PAGE _____

OF _____

SUBMITTED BY _____

ORGANIZATION _____

DATE _____

1	FUNCTION STATEMENT	72
\$CONVERT		

SEQUENCE
73 80

--

MODULE STATEMENTS

MODULE

SELECT STATEMENTS

SELECT

RENAME STATEMENTS

RENAME

CREATE STATEMENTS

CREATE

1 72

73 80

SUBMITTED BY _____

ORGANIZATION _____

DATE _____

		SEQUENCE	
1		73	80
	FUNCTION STATEMENT		
\$	REPORT		
	TITLE STATEMENT		
	TITLE		
	OPTIONS STATEMENT		
	OPTIONS		
	SELECT STATEMENTS		
	SELECT		
	OUTPUT STATEMENTS		
	OUTPUT		
1		73	80
		72	

INDEX

- ADD statement 4-2
- Alias
 - CONTROL category 3-2
 - Convert request 8-4
 - Entry 2-2
 - Query request
 - Count 5-3
 - List 5-3
 - Show 5-4
 - Selection for reports 6-3
- ALIAS category
 - Entry 2-2
 - Element entity 3-6
- ATTRIBUTES category
 - COBOL statement generation 7-5
 - Dataset entity 3-19
 - Element entity 3-5
 - File entity 3-14
 - Form entity 3-26
 - Group entity 3-8
 - Module entity 3-36
 - Record entity 3-10
 - Report entity 3-31
- Catalogue
 - Backup and restore 4-5
 - Definition 1-1
 - File characteristics 10-1
 - Maintenance 4-1
 - Production 9-2
 - Test 9-2
- Catalogue name
 - Conventions 2-1
 - Convert function 8-1
 - Range retrieval 5-10
 - RENAME statement 8-4
- Catalogue Report
 - Description 6-4
 - Example 11-1
 - OPTIONS statement 6-4
 - OUTPUT statement 6-4
 - Overview 6-1
 - \$REPORT statement 6-1
- Categories
 - ALIAS 3-6
 - ATTRIBUTES
 - Dataset entity 3-19
 - Element entity 3-5
 - File entity 3-14
 - Form entity 3-26
 - Group entity 3-8
 - Module entity 3-36
 - Record entity 3-10
 - Report entity 3-31
 - CLASSIFICATION 3-2
 - Common 3-1
 - CONTROL 3-2
 - DESCRIPTION 3-2
 - ENVIRONMENT 3-21
 - FLOW
 - Form entity 3-28
 - Report entity 3-31
 - IOAREA 3-24
- LOCATION 3-34
- NAMES
 - Data base entity 3-24
 - Dataset entity 3-18
 - Element entity 3-4
 - External resource entity 3-34
 - File entity 3-14
 - Form entity 3-26
 - Group entity 3-7
 - Manual task entity 3-41
 - Module entity 3-36
 - Record entity 3-10
 - Report entity 3-30
 - System entity 3-44
- OPERATING 3-37
- ORIGIN 3-3
- OTHER 3-2
- PASSING 3-39
- PERSONNEL 3-45
- RELATIONAL
 - External resource entity 3-34
 - Form entity 3-29
 - Manual task entity 3-43
 - Module entity 3-39
 - Report entity 3-32
 - System entity 3-45
 - User entity 3-46
- RESPONSIBILITY
 - Data base entity 3-23
 - Dataset entity 3-18
 - External resource entity 3-33
 - File entity 3-13
 - Form entity 3-25
 - Manual task entity 3-41
 - Module entity 3-35
 - Report entity 3-29
 - System entity 3-43
- SCHEDULING
 - Manual task entity 3-42
 - Module entity 3-38
- STORAGE 3-37
- STRUCTURE
 - Data base entity 3-24
 - Dataset entity 3-22
 - File entity 3-17
 - Group entity 3-9
 - Record entity 3-11
 - TOTAL record entity 3-12
- SUPPLIERS 3-26
- UNIT
 - Dataset entity 3-20
 - File entity 3-16
- VALUES 3-4
- Category
 - Header
 - Description 2-1
 - Statement 4-2
 - Line number
 - Description 2-1
 - Renumbering 9-2
 - Statement 4-4
- CLASSIFICATION category
 - Description 3-2
 - Keyword search 5-7

COBOL

- Convert function
 - Create Catalogue entries 8-2
 - Deck setup 8-3
- Generate function
 - Considerations 7-4
 - NAMES category 7-5
 - Source statement generation 7-1
- Level 88 definition 3-2
- Comment lines 2-2
- Common categories 3-1
- Continuation line
 - Category line 2-1
 - Function request 2-2
- CONTROL category 3-2
- Convert function
 - Applications 8-1
 - Checklisting data 8-1
 - Considerations 8-1
 - Control statements 10-8
 - CREATE statement 8-5
 - Description 8-1
 - Diagnostics B-18
 - Example 11-2
 - Executable programs 10-1
 - Field length 10-8
 - File usage 10-9
 - Module analysis 8-1
 - MODULE statement 8-3
 - Multiple record layouts 8-2
 - Operating requirements 10-8
 - Overview 1-5
 - Pseudo-switches 10-1
 - RENAME statement 8-4
 - Request structure 8-2
 - SELECT statement 8-4
 - \$CONVERT statement 8-3
- COUNT statement
 - Catalogue name range 5-10
 - Description 5-3
 - Entity type range 5-10
 - Search clauses 5-6
- CREATE statement 8-5
- CRMEP control statement 10-2

Data base entity

- Definition 1-2
- Description 3-23
- Field summary D-8
- IOAREA category 3-24
- NAMES category 3-24
- RESPONSIBILITY category 3-23
- STRUCTURE category 3-24

Data entities

- Aliases 2-2
- Data base 1-2, 3-23
- Dataset 1-2, 3-17
- Element 1-1, 3-3
- File 1-2, 3-13
- Group 1-1, 3-7
- Record 1-1, 3-10

Dataset entity

- ATTRIBUTES category 3-19
- Definition 1-2
- Description 3-17
- ENVIRONMENT category 3-21
- Field summary D-9
- NAMES category 3-18
- RESPONSIBILITY category 3-18
- STRUCTURE category 3-22
- UNIT category 3-20

Deck setups

- Convert function 8-3
- Generate function 7-1
- Query function 5-2
- Report function 6-1
- Update function 4-1

DEL statement 4-3

DESCRIPTION category

- Description 3-2
- Query request search 5-7

Element entity

- Alias 2-2
- ALIAS category 3-6
- ATTRIBUTES category 3-5
- Definition 1-1
- Description 3-3
- Field summary D-1
- NAMES category 3-4
- ORIGIN category 3-3
- VALUES category 3-4

Entity type

- Data base 3-23
- Dataset 3-17
- Element 3-3
- External resource 3-33
- Field summaries D-1
- File 3-13
- Form 3-25
- Group 3-7
- Hierarchy 5-9
- Manual task 3-40
- Module 3-35
- Range retrieval 5-10
- Record 3-10
- Report 3-29
- System 3-43
- User 3-45

Entry

- Category Header statement 4-3
- Category Line Number statement 4-4
- Change 4-2
- Comment line 2-2
- Convert function 8-1
- Creation 4-2, 8-1
- Definition 1-1
- Deletion 4-3
- Entity types 3-1
- Line renumbering 9-2
- Query request
 - Count 5-3
 - List 5-3
 - Show 5-4
- Range retrieval 5-10
- Renaming 9-2
- Selection for reports 6-2
- Structure 2-1
- Transfer 9-2

ENVIRONMENT category 3-21

Error processing

- CRM errors 10-2
- Diagnostics
 - Convert function B-18
 - Generate function B-15
 - Query function B-6
 - Report function B-9
 - Update function B-2
- Function identifiers B-1
- Input transactions B-1
- Overview 2-3
- Severity levels B-1
- Update function 4-5

External resource entity
Definition 1-2
Description 3-33
Field summary D-16
LOCATION category 3-34
NAMES category 3-34
RELATIONAL category 3-34
RESPONSIBILITY category 3-33

FILE control statement 10-2

File entity

ATTRIBUTES category 3-14
Definition 1-2
Description 3-13
Field summary D-6
NAMES category 3-14
RESPONSIBILITY category 3-13
STRUCTURE category 3-17
UNIT category 3-16

FLOW category

Form entity 3-28
Report entity 3-31

Form entity

ATTRIBUTES category 3-26
Definition 1-2
Description 3-25
Field summary D-12
FLOW category 3-28
NAMES category 3-26
RELATIONAL category 3-29
RESPONSIBILITY category 3-25
SUPPLIERS category 3-26

Functions

Convert

Description 8-1
Example 11-2
Overview 1-5

Generate

Description 7-1
Example 11-2
Overview 1-4

Query

Description 5-1
Example 11-1
Overview 1-3

Report

Description 6-1
Example 11-1
Overview 1-3

Request conventions 2-2

Update

Description 4-1
Example 11-1
Overview 1-3

Utility

Description 9-1
Overview 1-5

Generate function

Capabilities 7-1
COBOL considerations 7-4
Control statements 10-7
Description 7-1
Diagnostics B-15
Example 11-2
Executable programs 10-1
Field length 10-7
File usage 10-7
Operating requirements 10-7
OPTIONS statement 7-2
Overview 1-4

PL/I considerations 7-6
Pseudo-switches 10-1
Request output 7-1, 7-6
Request structure 7-1
SELECT statement 7-3
\$GENERATE statement 7-1

Group entity

Alias 2-2
ATTRIBUTES category 3-8
Definition 1-1
Description 3-7
Field summary D-3
NAMES category 3-7
STRUCTURE category 3-9

HAVING clause

Description 5-7
Logical connectors 5-9

Hierarchies

Entity types 1-2, 5-9
Query request 5-1
Report request 6-1

Hierarchy Report

Description 6-7
Example 11-2
OPTIONS statement 6-7
OUTPUT statement 6-4
Overview 6-1
\$REPORT statement 6-1

Index Report

Description 6-8
Example 11-2
OPTIONS statement 6-8
Overview 6-1
\$REPORT statement 6-1

IOAREA category 3-24

Keywords

CLASSIFICATION category 3-2
Query request search 5-7

LIST statement

Catalogue name range 5-10
Description 5-3
Entity type range 5-10
Search clauses 5-6

LOCATION category 3-34

Manual task entity

Definition 1-2
Description 3-40
Field summary D-20
NAMES category 3-41
RELATIONAL category 3-43
RESPONSIBILITY category 3-41
SCHEDULING category 3-42

Module entity

ATTRIBUTES category 3-36
Definition 1-2
Description 3-35
Field summary D-17
NAMES category 3-36
OPERATING category 3-37
PASSING category 3-39
RELATIONAL category 3-39
RESPONSIBILITY category 3-35
SCHEDULING category 3-38
STORAGE category 3-37
MODULE statement 8-3

- Name Analysis Report
 - Description 6-10
 - Example 11-2
 - OPTIONS statement 6-10
 - Overview 6-1
 - \$REPORT statement 6-1
- Names
 - Compound 6-10
 - Report 2-1, 3-1
 - System 2-1, 3-1
- NAMES category
 - COBOL statement generation 7-5
 - Data base entity 3-24
 - Dataset entity 3-18
 - Element entity 3-4
 - External entity 3-34
 - File entity 3-14
 - Form entity 3-26
 - Group entity 3-7
 - Manual task entity 3-41
 - Module entity 3-36
 - PL/I statement generation 7-6
 - Record entity 3-10
 - Report entity 3-30
 - System entity 3-44
- OPERATING category 3-37
- Operating requirements
 - Common files 10-1
 - Convert function 10-8
 - Executable programs 10-1
 - Generate function 10-7
 - Pseudo-switches 10-1
 - Query function 10-3
 - Report function 10-4
 - Update function 10-2
 - Utility function 10-2
- OPTIONS statement
 - Catalogue Report 6-4
 - Generate function 7-2
 - Hierarchy Report 6-7
 - Index Report 6-8
 - Name Analysis Report 6-10
 - Relational Report 6-9
 - Update function 4-2
 - Usage Report 6-5
- ORIGIN category 3-3
- OTHER category 3-2, 7-5
- OUTPUT statement 6-4
- Partial usage 2-2
- PASSING category 3-39
- PERSONNEL category 3-45
- PL/I
 - Generation considerations 7-6
 - NAMES category 7-6
 - Source statement generation 7-1
- Procedure entities
 - External resource 1-2, 3-33
 - Form 1-2, 3-25
 - Manual task 1-2, 3-40
 - Module 1-2, 3-35
 - Partial usage 2-2
 - Program 1-2, 3-35
 - Report 1-2, 3-29
 - System 1-2, 3-43
 - Versions 2-2
- Program entity (see also Module Entity)
 - Definition 1-2
 - Description 3-35
- Query function
 - Capabilities 5-1
 - Control statements 10-4
 - COUNT statement 5-3
 - Data administration 5-1
 - DESCRIPTION category search 5-7
 - Diagnostics B-6
 - Example 11-1
 - Executable programs 10-1
 - Field length 10-3
 - File usage 10-4
 - Information retrieval 5-1
 - Keyword search 5-7
 - Keywords 3-2
 - LIST statement 5-3
 - Operating requirements 10-3
 - Overview 1-3
 - Pseudo-switches 10-1
 - Range retrieval
 - Catalogue name 5-10
 - Entity type 5-10
 - Request structure 5-2
 - Search clauses 5-6
 - SHOW statement 5-4
 - TITLE statement 5-2
 - \$QUERY statement 5-2
- Record entity
 - Alias 2-2
 - ATTRIBUTES category 3-10
 - Definition 1-1
 - Description 3-10
 - Field summary D-5
 - NAMES category 3-10
 - STRUCTURE category 3-11
 - TOTAL STRUCTURE category 3-12
- RELATIONAL category
 - External resource entity 3-34
 - Form entity 3-29
 - Manual task entity 3-43
 - Module entity 3-39
 - Report entity 3-32
 - System entity 3-45
 - User entity 3-46
- Relational Report
 - Description 6-9
 - Example 11-2
 - OPTIONS statement 6-9
 - Overview 6-1
 - \$REPORT statement 6-1
- Relationships
 - Partial usage 2-2
 - Query request 5-1
 - Reporting 6-1, 6-9
 - Structure 1-2
- RENAME statement 8-4
- Report entity
 - ATTRIBUTES category 3-31
 - Definition 1-2
 - Description 3-29
 - Field summary D-14
 - FLOW category 3-31
 - NAMES category 3-30
 - RELATIONAL category 3-32
 - RESPONSIBILITY category 3-29
- Report function
 - Catalogue Report 6-4
 - Control statements 10-6
 - Description 6-1
 - Diagnostics B-9
 - Example 11-1

- Executable programs 10-1
- Field length 10-6
- File usage 10-5
- Hierarchy Report 6-7
- Index Report 6-8
- Keywords 3-2
- Name Analysis Report 6-10
- Operating requirements 10-4
- OPTIONS statement
 - Catalogue Report 6-4
 - Hierarchy Report 6-7
 - Index Report 6-8
 - Name Analysis Report 6-10
 - Relational Report 6-9
 - Usage Report 6-5
- OUTPUT statement 6-4
- Overview 1-3
- Pseudo-switches 10-1
- Relational Report 6-9
- Request structure 6-1
- SELECT statement 6-2
- TITLE statement 6-2
- Usage Report 6-5
- \$REPORT statement 6-1
- Report name
 - Definition 2-1
 - Field identification 3-1
 - Summary by entity type D-1
- Request forms
 - Samples E-1
 - Usage 2-3
- Request structure
 - Convert function 8-2
 - Generate function 7-1
 - Query function 5-2
 - Report function 6-1
 - Update function 4-1
- RESPONSIBILITY category
 - Data base entity 3-23
 - Dataset entity 3-18
 - External resource entity 3-33
 - File entity 3-13
 - Form entity 3-25
 - Manual task entity 3-41
 - Module entity 3-35
 - Report entity 3-29
 - System entity 3-43
- SCHEDULING category
 - Manual task entity 3-42
 - Module entity 3-38
- Screen entity (see Report Entity)
- Search clauses
 - HAVING 5-7
 - Logical connectors 5-9
 - USED-BY 5-8
 - WHICH-USE 5-9
 - WITH 5-6
- SELECT statement
 - Convert function 8-4
 - Generate function 7-3
 - Report function 6-2
- SHOW statement
 - Catalogue name range 5-10
 - Description 5-4
 - Entity type range 5-10
 - Search clauses 5-6
- Standard character sets A-1
- Statements
 - ADD 4-2
 - Category Header 4-3

- Category Line Number 4-4
- CHG 4-2
- COUNT 5-3
- CREATE 8-5
- DEL 4-3
- LIST 5-3
- MODULE 8-3
- OPTIONS
 - Catalogue Report 6-4
 - Generate function 7-2
 - Hierarchy Report 6-7
 - Index Report 6-8
 - Name Analysis Report 6-10
 - Relational Report 6-9
 - Update function 4-2
 - Usage Report 6-5
- OUTPUT 6-4
- RENAME 8-4
- SELECT
 - Convert function 8-4
 - Generate function 7-3
 - Report function 6-2
- SHOW 5-4
- TITLE
 - Query function 5-2
 - Report function 6-2
- \$CONVERT 8-3
- \$GENERATE 7-1
- \$QUERY 5-2
- \$REPORT 6-1
- \$UPDATE 4-1
- STORAGE category 3-37
- STRUCTURE category
 - Data base entity 3-24
 - Dataset entity 3-22
 - DBDL considerations 7-6
 - File entity 3-17
 - Group entity 3-9
 - Record entity 3-11
 - TOTAL record entity 3-12
- SUPPLIERS category 3-26
- System entity
 - Definition 1-2
 - Description 3-43
 - Field summary D-21
 - NAMES category 3-44
 - RELATIONAL category 3-45
 - RESPONSIBILITY category 3-43
- System name
 - Definition 2-1
 - Field identification 3-1
 - Index Report 6-8
 - Summary by entity type D-1
- Task entity (see Manual Task Entity)
- TITLE statement
 - Query function 5-2
 - Report function 6-2
- TOTAL
 - Convert function
 - Create Catalogue entries 8-2
 - Deck setup 8-3
 - Data base entity 1-2, 3-23
 - Dataset entity 1-2, 3-17
 - Entity hierarchy 1-2
 - Generate function 7-1, 7-6
 - Group entity 1-1
 - Record entity 1-2
 - Record structure 3-12

- UNIT category
 - Dataset entity 3-20
 - File entity 3-16
- Update function
 - ADD statement 4-2
 - Audit trail 4-5
 - Category Header statement 4-3
 - Category Line Number statement 4-4
 - CHG statement 4-2
 - Control statements 10-3
 - Convert function output 8-1
 - Deck setup 4-1
 - Description 4-1
 - Diagnostics 8-2
 - Error processing 4-5
 - Examples 11-1
 - Executable programs 10-1
 - Field length 10-2
 - File usage 10-2
 - Logging 4-5
 - Operating requirements 10-2
 - OPTIONS statement 4-2
 - Overview 1-3
 - Pseudo-switches 10-1
 - Request structure 4-1
 - \$UPDATE statement 4-1
- Usage Report
 - Description 6-5
 - Example 11-2
 - OPTIONS statement 6-5
 - OUTPUT statement 6-4
 - Overview 6-1
 - \$REPORT statement 6-1
- USED-BY clause 5-8
- User entity
 - Definition 1-2
 - Description 3-45
 - Field summary D-22
 - Partial usage 2-2
 - PERSONNEL category 3-45
- RELATIONAL category 3-46
 - Versions 2-2
- Utility function
 - Backup and restore 9-2
 - Definition transfer 9-2
 - Executable programs 10-1
 - Footnote message 9-1
 - Installation identification 9-1
 - Line renumbering 9-2
 - Overview 1-5
 - Page overflow 9-1
 - Renaming entries 9-2
 - Required fields 9-1
 - System initialization 9-1
 - System statistics 9-2
- VALUES category 3-4
 - Version
 - CONTROL category 3-2
 - Convert request 8-4
 - Entry 2-2
 - Query request
 - Count 5-3
 - List 5-3
 - Show 5-4
 - Selection for reports 6-3
- WHICH-USE clause 5-9
- WITH clause
 - Description 5-6
 - Logical connectors 5-9
- \$CONVERT statement 8-3
- \$GENERATE statement 7-1
- \$QUERY statement 5-2
- \$REPORT statement 6-1
- \$UPDATE statement 4-1

COMMENT SHEET

MANUAL TITLE: Data Catalogue 2 Reference Manual

PUBLICATION NO.: 60483200

REVISION: A

NAME: _____

COMPANY: _____

STREET ADDRESS: _____

CITY: _____ **STATE:** _____ **ZIP CODE:** _____

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

CUT ALONG LINE

AA3419 REV. 4/79 PRINTED IN U.S.A.

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

FOLD ON DOTTED LINES AND STAPLE

OLD

FOLD



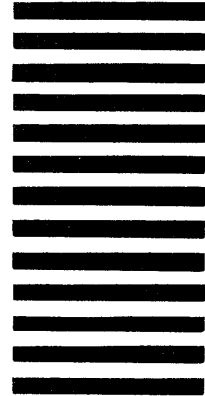
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION

Publications and Graphics Division
215 Moffett Park Drive
Sunnyvale, California 94086



CUT ALONG LINE

OLD

FOLD

CORPORATE HEADQUARTERS, P.O. BOX 0, MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.



CONTROL DATA CORPORATION

102680408