IRM-5235-01A

U.S. Marine Corps

DATA DICTIONARY

PCN 186 523501 00

DEPARTMENT OF THE NAVY
HEADQUARTERS UNITED STATES MARINE CORPS
WASHINGTON, D.C. 20380-0001

IN REPLY REFER TO:
5235/01A
CTAS-31
3 Feb 1992

From: Commandant of the Marine Corps

Subj: INFORMATION RESOURCES MANAGEMENT (IRM) DATA DICTIONARY

Ref: (a) MCO 1510.37
(b) MCO P5231.1
(c) MCO 5271.1
(d) MCO P5600.31

Encl: (1) IRM-5235-01A

1. PURPOSE. To provide guidance and instructions on the development of Data Dictionaries as required by references (a) and (b).

2. CANCELLATION. IRM-5235-01.

3. SUMMARY OF REVISION. This revision updates data dictionary terminology and changes all references to DESIGNMANAGER to CONTROLMANAGER/DATAMANAGER. A glossary has been added at Appendix J. The Department of Defense (DoD) Corporate Information Management Act (CIM) has mandated significant changes in the existing Data Administration programs. The CIM initiative will ensure the standardization and consistency of data from the DoD's information systems. Updates to the existing Data Dictionary technical publication will be provided as they are received.

4. AUTHORITY. This publication is published under the auspices of reference (c).

5. APPLICABILITY. The guidance contained in this publication is applicable to all contractors and Marine Corps personnel responsible for the preparation of a data dictionary. This standard is applicable to the Marine Corps Reserve.

6. DISTRIBUTION. This technical publication will be distributed as indicated. Appropriate activities will receive updated individual activity Table of Allowances for Publications. Requests for changes in allowance should be submitted in accordance with reference (d).

7. SCOPE

a. Compliance. Compliance with the provisions of this publication is required unless a specific waiver is authorized.

b. Waivers. Waivers to the provisions of this publication will be authorized only by CMC (MCCTA) on a case by case basis.

Subj: INFORMATION RESOURCES MANAGEMENT (IRM) DATA DICTIONARY

8. SPONSOR. The sponsor of the technical publication is CMC (MCCTA).

D. M. BLACK
By direction

DISTRIBUTION: PCN 186 523501 00

Copy to: 8145001

UNITED STATES MARINE CORPS

Information Resources Management (IRM) Standards and Guidelines Program

Data Dictionary
IRM-5235-01A

Enclosure (1)

(This page intentionally left blank)

TECHNICAL PUBLICATION LIBRARY MAINTENANCE

The Information Resources Management Standards and Guidelines Program publications will be maintained at each receiving activity. Each activity is responsible for ensuring that their set of technical publications is complete, and that all published changes are promptly incorporated.

RECORD OF CHANGES

Change Number	Date of Change	Date Received	Date Entered	Signature of Person Entering Change

(This page intentionally left blank)

PUBLICATION TABLE OF CONTENTS

(This page intentionally left blank)

LIST OF FIGURES

(This page intentionally left blank)

Chapter Table of Contents

Chapter 1

GENERAL

(This page intentionally left blank)

Chapter 1

GENERAL

1.1. INTRODUCTION. The DATAMANAGER Data Dictionary is intended to be the central repository of the descriptions of each system. It should accurately define the process entities and the data entities.

1.1.1. Benefits. By defining and controlling the parts that constitute each system with a data dictionary, the following benefits are realized:

a. Data entities are defined in one place, eliminating inconsistent definitions of the same entity. Also, standards can be applied across all development and support teams.

b. Relationships may be established between the data dictionary entities. This allows the accurate analysis of the impact of a change made to particular data or a process entity. For example, if a data field size is to be changed, the data dictionary should identify all include members that must be changed and which programs must be recompiled.

c. Data entity descriptions used by the programs may be generated by the data dictionary.

d. Documentation generated from the data dictionary can be more accurate since it reflects the descriptions actually used in the systems.

1.1.2. Requirements. The benefits of a data dictionary are accrued only if the following requirements are met:

a. The data dictionary contains accurate, complete information about each system.

b. The information is available in a format that is useful to those that use the data dictionary.

c. The update or retrieval procedures are not overly burdensome to the point that the costs of maintaining it outweigh the benefits.

1.2. SCOPE.

1.2.1. Issues. This standard addresses issues concerning the usage and application of the data dictionary, operation in the data dictionary environment, definition of responsibility for administrative and technical management, and conventions for use of data dictionary facilities. Documentation in the data dictionary includes the components of the data flow diagrams (DFD's) used to specify the design.

1.2.2. Applicability. This standard is intended to apply to DATAMANAGER. The information that is to be maintained on the data dictionary is a subset of that delivered as the formal design documentation. This includes components of the Functional Requirements Definition (FRD), the General Design Specification (GDS), and the Detailed Design Specification (DDS). This standard addresses application program and data base design documentation.

1.3. APPROACH. The following paragraphs describe the approach to be taken when using the data dictionary. Topics addressed describe the structure, control, and relationships of the data dictionary.

1.3.1. Dictionary Usage. Multiple versions of the dictionary should be used to reflect different stages of the development effort (that is, FRD, GDS, DDS). Each version is comprised of an unlimited number of members (dictionary entries).

1.3.2. Documenting Applications. Since the top down approach will be utilized in the development cycle, the data dictionary should reflect this same approach. DATAMANAGER will be used to document applications in the same way as they are defined during design by entering one system data definition per application. Figure 1-01, "Top Down Approach," illustrates this point. This view of the member hierarchy shows the minimum documentation required in the data dictionary. Use of nesting within any of these member types is expected to be used at each developer's discretion to further define their design according to their own documentation requirements. Also, the top down approach can be utilized with a bottom up data entry. In this manner data items identified in the data requirements document could be entered, thus facilitating redundancy searches and data modeling. The information used to describe and link members is listed in Figure 1-02, "Required Information."

1.3.3. Multiple Physical Dictionaries. DATAMANAGER is not restricted to handling a single physical data dictionary. Facilities are provided for the transfer of entries between separate dictionaries. These facilities should be used to help in the control of multiple dictionary versions in the development cycle. This is very useful as a management tool and should be used accordingly. These facilities are among those provided for the overall control, reorganization, and security of data dictionaries, and are described in the DATAMANAGER Controller's Manual. The Data Administrator will establish a single Production Dictionary, and multiple Project Dictionaries (one for each development project). The use of the "status" feature will be employed where possible to minimize the number of physical dictionaries. There should be only a single physical dictionary per site.

1.3.4. Multiple Logical Dictionaries. The DATAMANAGER Status feature allows the construction of multiple logical dictionaries within one physical dictionary and should be used in both the development and maintenance environments. This enables users to achieve the following two basic objectives:

a. It allows the recording of data definitions and associated relationships held in the dictionary at different points in time. This provides historical and additional backup documentation.

b. It allows the recording of different data definitions and/or different relationships in the dictionary based upon different views of the data. This provides for the representation of various data analysis methodologies (for example; conceptual and implementation, logical and physical).

PROCESS MEMBERS

DATA MEMBERS

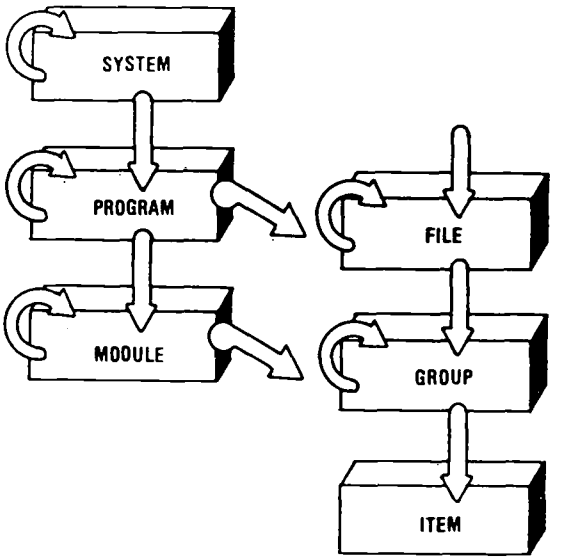


FIGURE 1-01
Top Down Approach

DATAMANAGER CLAUSES	MEMBER TYPES: SYSTEM	PROGRAM	MODULE	FILE	GROUP	ITEM
Inputs	X	X	X			
Outputs	X	X	X			
Updates	X	X	X			
Contains	X	X	X	X	X	
Catalogue	X	X	X			
Calls		X	X			
Parameter		X	X			
Entered-As				X	X	X
Held-As				X	X	X
Reported-As				X	X	X

FIGURE 1-02
Required Information

(This page intentionally left blank)

Chapter Table of Contents

Chapter 2

OPERATION AND USE

(This page intentionally left blank)

Chapter 2

OPERATION AND USE

2.1. INTRODUCTION. This section describes the operation and usage characteristics of data dictionary members. The discussion describes the structure of data dictionary, the type, control and relationship to one another, and responsibilities for data contained therein. Also included are sub-sections on logical modeling and reporting features of DATAMANAGER.

2.1.1. Dictionary Features. If sufficiently justified, certain features may require restrictions for use to ensure integrity. As restrictions are identified, they will be added to each of the sub-paragraphs below.

a. Source Language Generation. This facility allows application software access to definitions, descriptions, and structure directly from the data dictionary. The automatic generation of source code (COBOL File Definition, ADABAS, etc.) is from structures identified in DATAMANAGER. Three manuals contain information about the use of this facility: DATAMANAGER User's Guide, DATAMANAGER Source Language Generation Manual, and DATAMANAGER (ADABAS) Interface Manual.

b. Automation of Set Up. DATAMANAGER can generate data dictionary definitions from existing application source code. For more information see the "Automation of Set Up Manual".

c. User Interface Feature. The DATAMANAGER User Interface Feature provides a means by which a DATAMANAGER Data Dictionary can be interfaced with user-written software. It allows the user to input data through the use of interactive screens or other "front ends", and also permits the user to output DATAMANAGER information; consult the DATAMANAGER "User Interface Facility Manual".

d. Relationships to other Data Dictionaries. By definition, DATAMANAGER is a free standing data dictionary. This implies that its use is indirectly related to any given suite of Data Base Management System (DBMS) software to include any DBMS internally driven DD. It is the responsibility of the functional user to ensure that the usage and constraints are compatible and synchronized.

e. Interface to ADABAS. This facility of DATAMANAGER will produce data definitions in compressed format from a predefined DD member type of ADABAS-FILE. Output from this facility will serve as input for procedures to create and maintain all required ADABAS structures necessary for ADABAS file definitions and access. Such access includes the ability to access NATURAL via data definition modules (DDM).

f. Logical Modeling. Logical modeling is a facility of the data dictionary. This feature analyzes the relationship among the data elements and provides graphic displays and reports using a normalization process. These outputs assist the data base designers in determining subject data bases.

2.1.2. Responsibilities. Each DATAMANAGER Controller or System Administrator must provide a user's manual that describes responsibilities for the data dictionaries under his/her control.

2.1.3. Multiple Dictionaries.

a. The DATAMANAGER status feature. The status feature allows the construction of multiple logical dictionaries within one physical dictionary. This allows:

(1) the recording of how data definitions and associated relationships are held in the dictionary at different points in time; and

(2) the registration of different data definitions and/or different relationships in the dictionary based upon different view of the data. The status feature can be used to undertake development of new relationships and structures, or modify existing relationships and structures, while the basic structure exists for production purposes.

b. Detailed information. Detailed information on the status feature and its use is contained in the DATAMANAGER "Introduction to Status".

c. Control. Use of the status feature will be controlled by the DATAMANAGER Controller. Unless authorized or assigned by the Controller, all users will use the default status.

2.1.4. Data Definition.

a. Methodology. Refer to Appendix A for the discussion that describes the methodology of creating names and abbreviations to assure common terminology and consistency in naming conventions.

b. Formats, structures, and conventions. Formats, structures and conventions in representing common data elements such as dates and numerical values are found in Appendix B.

c. Minimum Descriptive Information. Definition of minimum descriptive information that must be present for each data dictionary entry will be found in Appendix C.

2.1.5. Member Types. Appendix D defines the authorized dictionary member types available with User Defined Syntax (UDS). All dictionary member types are based on, and acquire the same

attributes of one of the DATAMANAGER basic member types. (The basic member types are System, Program, Module for Process Definitions and File, Group, Item for Data Definitions.) A database member type is a special case that has attributes of both process definitions and data definitions. Any deviations from the authorized dictionary member types must be requested by the user via the supporting MCCDPA and approved by the Data Administration Officer CMC (CTAE).

2.1.6. Member Control. The appropriate user's manual will describe the procedures to maintain and modify data dictionary members.

2.1.7. Use of Member types. Appendix E provides detailed instructions for the use of DATAMANAGER member types.

2.1.8. Adding Members to a Dictionary.

a. General. One of the most difficult issues in data dictionary use is the assignment of names to data elements or any member in the dictionary. If names are assigned at random, with no logical method of construction, the results are a chaotic, disorganized system. The 32 character limitation on member name length internal to DATAMANAGER is an additional consideration. These problems can be alleviated by establishing standard methods for building member names. However, generating member names is normally performed at the time that a member is added to a dictionary. Therefore, a standard method of adding members must be provided along with the standard method of devising a member name.

b. Procedure for Adding Members. The following paragraphs show the procedure to be used when adding a member to a DATAMANAGER dictionary. These procedures ensure that consistency is maintained within the dictionary, and will result in a completed member work sheet for a member with a unique name. This method of constructing member names will minimize the possibility of naming conflicts within a dictionary. These procedures are set forth here to ensure uniformity throughout the Marine Corps.

c. Define Member. Before a member can be added to the dictionary, it must be clearly defined. In fact, it usually cannot be named before a definition is written. Use the appropriate member work sheet (Appendix F) to record the definition.

2.1.9. Member Relationship.

a. General. The purpose of defining members is to identify them in such detail as to allow the analyst to easily determine their function and their relationship to other members.

Identification of relationships facilitates operations, systems analysis and system design.

b. Identification of Users of Applications, Systems and Reports. One issue which DATAMANAGER will assist in resolving is the identification of users of applications, systems and reports. By indicating which members a user has need of, a comprehensive picture of information requirements can be discerned. User relationships are indicated by making a CONTAINS entry in the user's ORGANIZATION member indicating that a particular member is used by that member. Where a user employs a whole system the CONTAINS clause need only refer to the system, and need not indicate all of its components. CONTAINS statements should reference the highest level at which a user has a need for data (e.g. the report level, or the application level). NEVER MAKE BOTTOM-UP REFERENCES. All references must be from a higher to a lower member.

c. Identification of Functional Structures. A second shortcoming of current documentation is that there is no easy way to identify the applications and systems which collectively allow a particular function to be accomplished. It is therefore valuable to identify functional structures to DATAMANAGER. A CONTAINS clause should be placed in the function members, indicating the members which perform those functions. As usual, the relationships are hierarchical, so that a relationship used only refers to the highest level member which performs the function.

d. Indicating Responsibility for Data. It is essential that responsibility for the accuracy of data be established. Responsibility is established at the ITEM level. Identification is accomplished by making a MAINTAINED "member" CATALOG entry in the ITEM member indicating the office responsible for that particular element of data. The member name entered in the CATALOG statement is the name of the organizational member responsible for the description of the data element. The Controller will refer all questions about the characteristics of an element to the entity responsible for it. All data being entered into the dictionary will be assigned to a Functional Manager. The sources of the data and primary user also will be catalogued.

2.1.10. Line Numbering Conventions. If line numbering conventions are used for DATAMANAGER, Appendix G applies.

2.1.11. Data Flow Diagrams. Appendix H describes the use of DATAMANAGER for storing Data Flow Diagrams.

2.1.12. DATAMANAGER Reporting. DATAMANAGER provides extensive reporting capabilities on information stored within the data dictionary. The reporting facility is capable of producing

voluminous reports and therefore the danger of misuse exists. Listed below are typical reports that may be utilized.

a. DATAMANAGER documentation commands:

(1) Lists of member names, catalogues, aliases according to a variety of selection criteria.

(2) Printouts of contents of member definitions.

(3) Reports of member definitions plus related definitions.

(4) Glossary of members that shows only selected parts of the definitions.

b. DATAMANAGER interrogation commands

(1) Does one member use another.

(2) What/which members use or are used by other members.

(3) What/which members are catalogued.

(4) Whose alias is catalogued.

2.2. DOCUMENTATION DEPENDENCIES. The documentation governed by this standard may also rely on the content of other deliverables and/or standards.

2.2.1. Change Requirements. Since the SDM is an integrated methodology, there exists a relationship between documents in that preceding documents provide information to the follow-on documents. During the development of the Data Dictionary new issues may arise that will require changes to preceding documents. These changes must be documented and approved in accordance with the Quality Assurance and Configuration Management procedures. Externally imposed milestones that are unrealistic to accomplish should not be used as an excuse to defer or eliminate the documentation requirements.

(This page intentionally left blank)

Appendix A

NAMING MEMBERS

Member names should always be as generic as possible while still describing that entity being defined. Member names should not exceed 26 characters to meet the ADABAS considerations and to stay within the constraints of COBOL. The following describes the two broad categories of dictionary members and the naming conventions that apply:

(1) Members that define the real, physical data that forms the heart of the corporate data resource. These are usually the item, group and normalized DBMS file definitions. These member names should never indicate a specific system, process or context in which they are used. For example, the definition for social security number should be simply SOCIAL-SECURITY-NUMBER, not JUMPS-SOCIAL-SECURITY-NUMBER. The use of acronyms in the member names should always be avoided.

(2) The second category of members defines everything else: That is to say the organizational and analysis/design hierarchy members, as well as the process members of the data processing hierarchy. These names should also be as generic as possible. However, these members often pertain to specific systems or functions. Consequently, the member name is permitted to contain references to specific systems or functions. For instance, a process member that defines the JUMPS/MMS process might be called JUMPS-UPDATE-AND-EXTRACT. A generalized process to validate dates might be called simply DATE-VALIDATION (the use of generic names in this situation is particularly necessary if the date validation process is used by many different systems).

Avoid redundancy in member names. In other words, do not include references to the member type in member name. For example, the file definition for the JUMPS/MMS Master File should be called JUMPS-MASTER; not JUMPS-MASTER-FILE. The member type can be a criteria for any DATAMANAGER interrogation and will be obvious on any DATAMANAGER output reports.

Form Initial Name. Once a member has been defined, a member name can be constructed by utilizing all operative business words. Omit prepositions and conjunctions. Words should be separated by hyphens. In the case of members such as programs and procedures, where the member is used exclusively by the CDPA, the step or procedure name (e.g. I4524224) can be used, but an alias clause should identify the purpose of the member.

Form First Abbreviated Name. Abbreviate all words in the name using the following steps.

Check to see if there is an abbreviation in the glossary.

WHAT FORMS "FULL-SPELLING WORD";

If there is a response, then the member name which results is the abbreviation to be used in the abbreviated title.

If no member is identified, then fill out a Glossary Member Work Sheet, with the word to be abbreviated as the CATALOG entry "FULL-SPELLING word", and the abbreviation as the member name. Abbreviations can be extracted from relevant reference documents, such as the CODES Manual, or other authoritative functional area references. The meaning of the term should be entered in the DESCRIPTION of the glossary member. All other information indicated on the sheet must also be provided.

Abbreviate the word in the title.

Continue until all abbreviations have been provided.

Catalog all Abbreviations. Make catalog entries on the member work sheet for all abbreviations now in the member name. This will permit a user to retrieve the member even if the abbreviation is deleted from the member name later on.

Check Whether the Item is Already in the Dictionary. Use a WHAT FORMS "word" command to see what members use each of the abbreviations identified in the member name. Use this list to determine whether the member being defined is already in the dictionary.

Shorten Abbreviated Name. Shorten the member name to 26 characters by deleting the most general abbreviations. This will permit the member name to be used in COBOL programs, and will reduce the number of characters DATAMANAGER users will have to type to access the member.

Add Member to Dictionary. When the member has been fully described and named, it may be entered to the dictionary. DA will establish local procedures for this, but it is imperative that tight control be exercised by the Data Base Controller (DBC) to ensure that the dictionary does not become populated by incomplete or unused members. The DBC will maintain the glossary, and ensure that necessary abbreviations are added to it.

When adding a member to a dictionary, it is recommended that the ADD command be used, and that the member and its member type by the only information entered at first. An ADD command will encode automatically, and entry of only the member type will prevent a large amount of typing from being lost if the member already exists in the dictionary.

Adding Members in Batch Mode. Use of batch processing is the preferred method of updating the dictionary when more than two

individuals are using the dictionary simultaneously. When DATAMANAGER performs updates, it locks out all inquiries. This results in users being unable to interrogate the dictionary for the time it takes for the update to be completed. This is complicated by the fact that updates always take precedence over inquiries. It is possible for user inquiries to wait indefinitely while queued updates are performed.

(This page intentionally left blank)

Appendix B

DATA ELEMENT REPRESENTATIONS

Date Formats. Although there are multiple variations of dates for display purposes, there are two legitimate date formats for DATAMANAGER: Julian Date and Gregorian Date. Julian Date is in the format YYNNN, where YY is the last two digits of the year, and NNN is the number of the day in the year. February 15 1992 is represented as 92046. The other acceptable date format is the form YYMMDD, where YY is the last two digits of the year, MM is the number of the month, and DD is the number of the day. February 15 1992 is represented as 920215. Other date formats must be approved by the Data Administration Officer CMC (CTAE) before they can be used.

Calculated Fields. Calculated fields, fields which are the result of performing an arithmetic operation on one or more fields, must be identified in DATAMANAGER any time that the calculated value is stored in a file. The equation used to derive the value must be stored in the NOTE clause of the member description.

(This page intentionally left blank)

Appendix C

MINIMUM DESCRIPTIVE INFORMATION

General. The purpose of DATAMANAGER is to capture all information pertaining to a member. However, it is usual for only partial information to be available at the time that a member must first be entered into the dictionary. This minimum information represents the minimum information required for the member to be identified within the context of other members. Without the minimum information, the item shall not be entered into the dictionary. Additional information regarding the individual clauses which constitute minimum required information follows.

Minimum Information Required For Process Members. The following entries are mandatory for all process members:

Member Type	(ORGANIZATION, FUNCTION, USER, EXTERNAL-PROCESS, PROCESS, SUBPROCESS, APPLICATION, SYSTEM, JOB, PROCEDURE, TRANSACTION, JOB-STEP, PROGRAM, MODULE, SUBROUTINE, or DATASTORE)
Description	(Minimum of Long Title and brief description)

Minimum Information Required for Data Members. The following entries are mandatory for all data members except ITEMS and ELEMENTS:

Member Type	(FILE, LOGICAL-FILE, PHYSICAL-FILE, DATASET, REPORT, DOCUMENT, SCREEN, FORM, RECORD, DATAFLOW, DATASTRUCTURE, OR GROUP)
DESCRIPTION	(Minimum of Long Title and brief description.

Minimum Information Required for ITEMS. The following entries are mandatory for ITEMS:

ITEM
DESCRIPTION (Minimum of Long Title and brief description)

One of the following form descriptions:
- ENTERED-AS
- HELD-AS
- REPORTED-AS
- DEFAULTED-AS

Minimum Information Required for COMMAND-STREAMS. The following entries are mandatory for all COMMAND-STREAMS:

```
COMMAND-STREAM
DESCRIPTION        (Minimum of Long Title and brief
                    description)
CONTENTS
```

Minimum Information Required for ENTRY). The following entries are mandatory for all Glossary members:

```
ENTRY
DESCRIPTION        (Minimum of Long Title and brief
                    description)
ALIAS "FULL-SPELLING=full spelling" where full spelling
                    is the full spelling of the abbreviation
                    for which the entry is being made.
CATALOG "STANDARD ABBREVIATION"
```

Appendix D

AUTHORIZED DICTIONARY MEMBER TYPES

SYSTEM Members

ORGANIZATION	The administrative structure which performs assigns functions but does not directly use a system.
FUNCTION	A discrete action taken by an organization to achieve a given goal.
DEPARTMENT	The largest work unit contained within an organization.
DIVISION	The part of an organization that performs a major task. Usually operates within a department.
EXTERNAL-PROCESS	A process which occurs outside of a system being defined. Data Flow Diagram: A data terminator/originator. Normally used in context schemes.
PROCESS	A process changes incoming flows into outgoing flows. Data Flow Diagram: The circle in a data flow diagram to represent a process.
BRANCH	The next work unit breakdown for a division. Subordinate to a division and superior to a section.
SUBPROCESS	A sub-process is used in the fragmentation or decomposition of a process. Contains the same attributes as a process.
SECTION	The smallest organizational entity that does work. This member type can also be used to represent a unit, team or task group.
SYSTEM	A collection of men, machines, and methods organized to accomplish a set of specific data processing functions.
USER	The action officer or action section which performs a specific function or task.

PROGRAM Members

JOB	A specified group of tasks prescribed as a unit of work for a computer. By extension a job usually includes all necessary computer programs, linkages, files, and instructions to the computer operating system. A secondary or subordinate system.
PROCEDURE	The course of action taken for the solution of a single problem. A collection of PROGRAM, MODULES and SUBROUTINEs to do a specific task.
TRANSACTION	A grouping of data elements used to perform an update of a file.
JOB-STEP	In OS, the JCL instructions to execute a program and specify which files to use.
PROGRAM	A series of instruction or statements acceptable to a computer. A collection of instructions, modules and subroutines organized and linked together to form a method for problem solving.

MODULE Members

MODULE	A discrete identifiable set of instructions, organized to perform a specific task.
SUBROUTINE	The smallest collection of code designed to do the most basic tasks.
DATASTORE	Data Flow Diagram: A data store is represented by two parallel lines in a data flow diagram. It is used to define data at rest that is changed by the process accessing it.

ADABAS-DATABASE

Collection of ADABASE files used to form a database. This member type has attributes of both process definition members and data definition members.

FILE Members

FILE	A group of records which together form a structure that supports the requirements of physical or logical views.

LOGICAL-FILE	A logical grouping of information which supports a specific need. A logical file does not take into consideration system constraints on file design.
PHYSICAL-FILE	A file which exists in physical form. A physical file must meet all requirements and restrictions imposed by the system or environment in which it exists. A physical file exists on one or more datasets.
DATASET	A discrete grouping of data identified by a unique dataset name (DSN).
REPORT	A formatted extract from one or more files which will provide information in a form usable by the person requesting the report.
DOCUMENT	A written, typed or printed representation of stored information.
ADABAS-FILE	Collection of ADABAS field names (Data Items), including descriptors, subdescriptors, and super descriptors.

GROUP-Members

SCREEN	An interactive means of providing information to or from a database or file(s) via a terminal (CRT). Screens are developed in support of processes which allow the user to inquire, modify, add, or delete information on a database or file.
FORM	A preformatted arrangement of reserved areas for the insertion of data.
RECORD	A grouping of one or more items or groups of items which is the largest component of a file, logical file, or physical file.
DATAFLOW	A data flow is composed of data structures. It is an instance of data in motion, between a data store and a process or a file member-type and a process. A data flow to a data store can represent: - putting new data structures into the data store - updating data structures in a data store - accessing data structures within a data store

DATASTRUCTURE	A logical grouping of data which moves between data stores, or is required by a process in order to do its work. A data flow may contain one or more data structures. This is synonymous to a packet in Yourdon.
GROUP	A grouping of two or more items which have individual meanings, but which also have a unique meaning when associated.
ENTITY	Something about which data is recorded.

ITEM Number

ITEM	A discrete datum which has a unique meaning and cannot be divided into other data which also have unique meanings. The most detailed item identified in DATAMANAGER.
ELEMENT	A member used exclusively to record abbreviations used in member names. It is used to ensure that standard abbreviations are used in construction of member names. A glossary member is named with the abbreviation and contains one or more alias clauses indicating the words which are replaced by that abbreviation.
ENTITY-ATTRIBUTE	Contains information about an entity.

COMMAND-STREAM Member

COMMAND-STREAM	A series of DATAMANAGER commands which can be invoked by utilizing the PERFORM command.

Appendix E

DATAMANAGER MEMBER TYPES

General. There are three general member types in DATAMANAGER, two of which describe data elements and structures, and a third which contains commands which can be executed as a whole in lieu of keying in commands in the interactive mode.

These three basic member types do not permit the clear delineation of functional entities within either the process or data structures. The User Defined Syntax (UDS) feature permits more functional, flexible and diverse identification of DATAMANAGER members. The standard dictionary structure within the Marine Corps will be the DATAMANAGER Structured Development Structure (SDS) shown in Figure E-1. Detailed instructions for its use are contained below. Additional information may be found in the DATAMANAGER USER DEFINED SYNTAX MANUAL and the DATAMANAGER USER'S GUIDE.

Member Types in DATAMANAGER. DATAMANAGER employs a structure of members to construct representations of systems and data elements. These structures are then linked to represent relationships inherent in the systems being defined. The two types of members used to describe systems in DATAMANAGER are process and data member types.

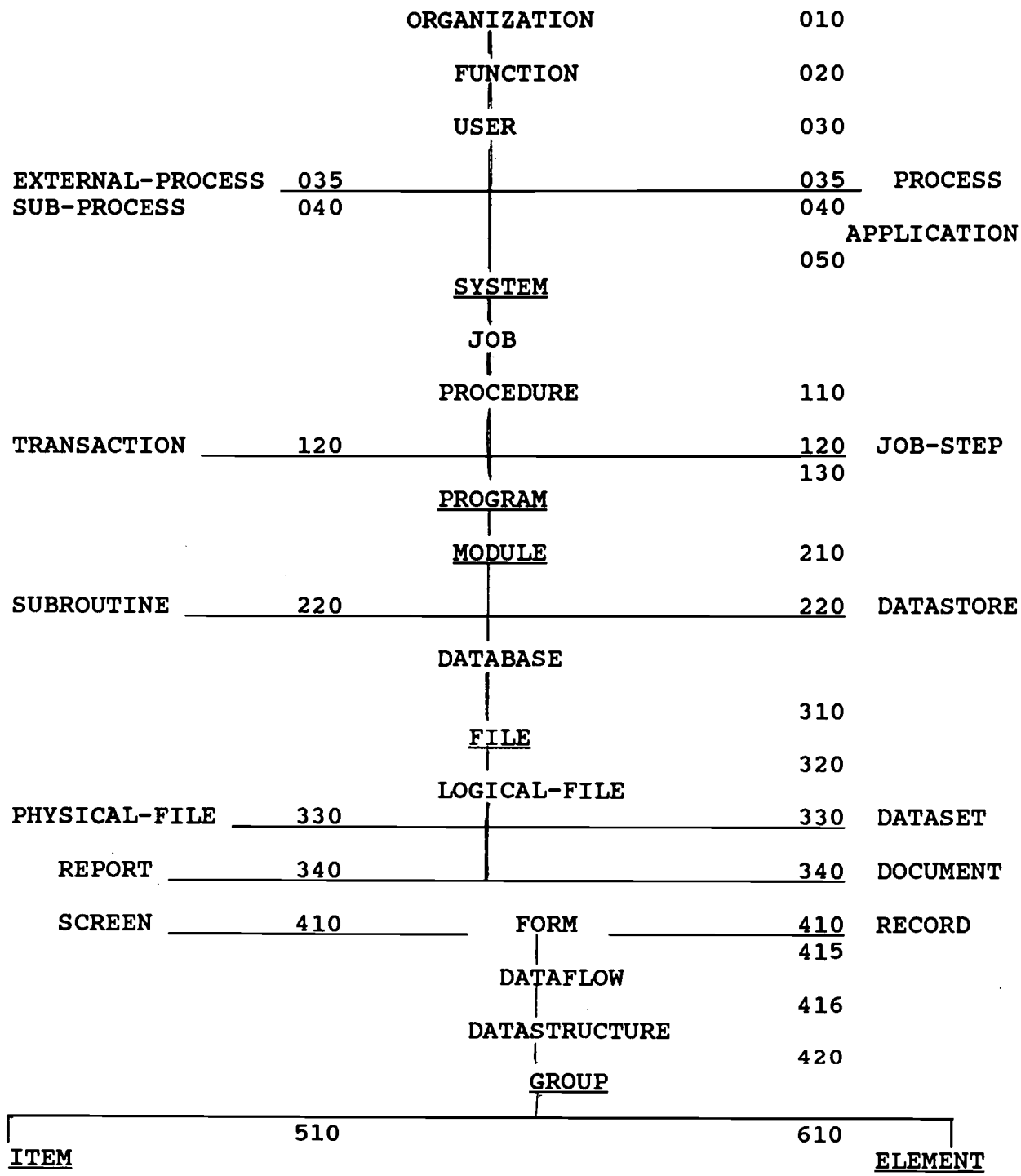
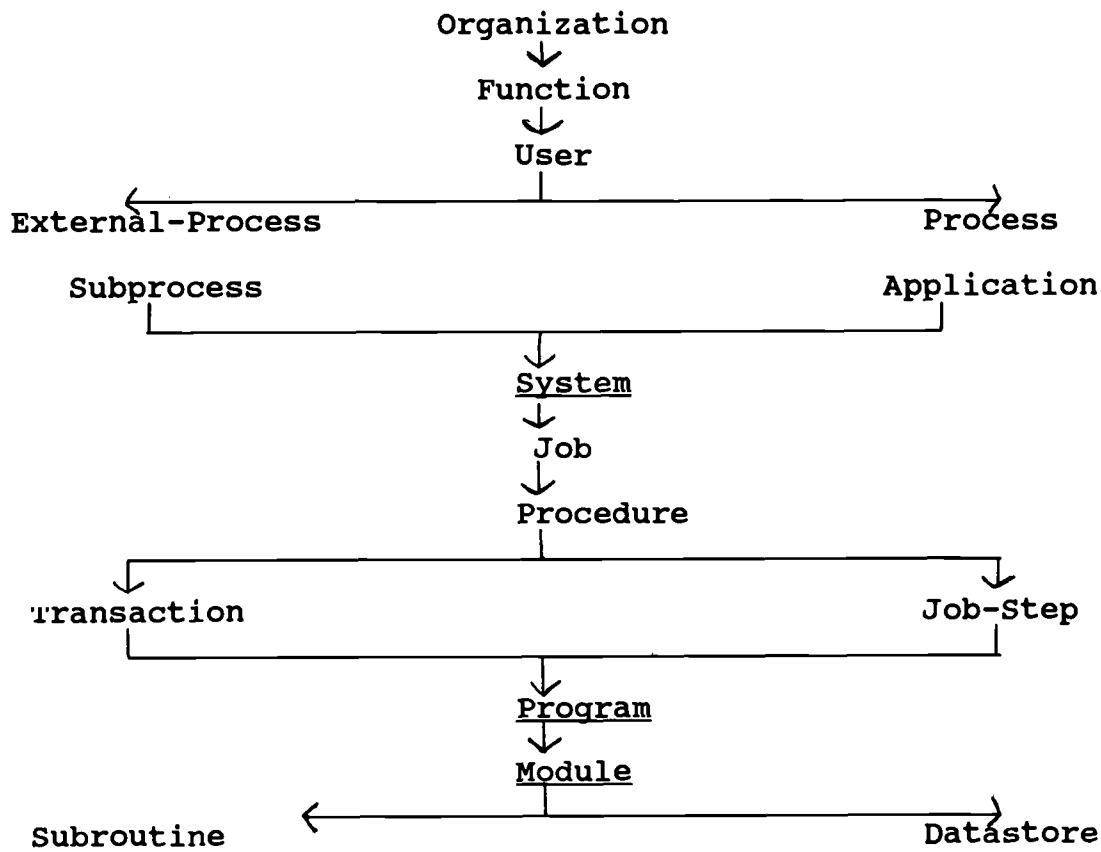
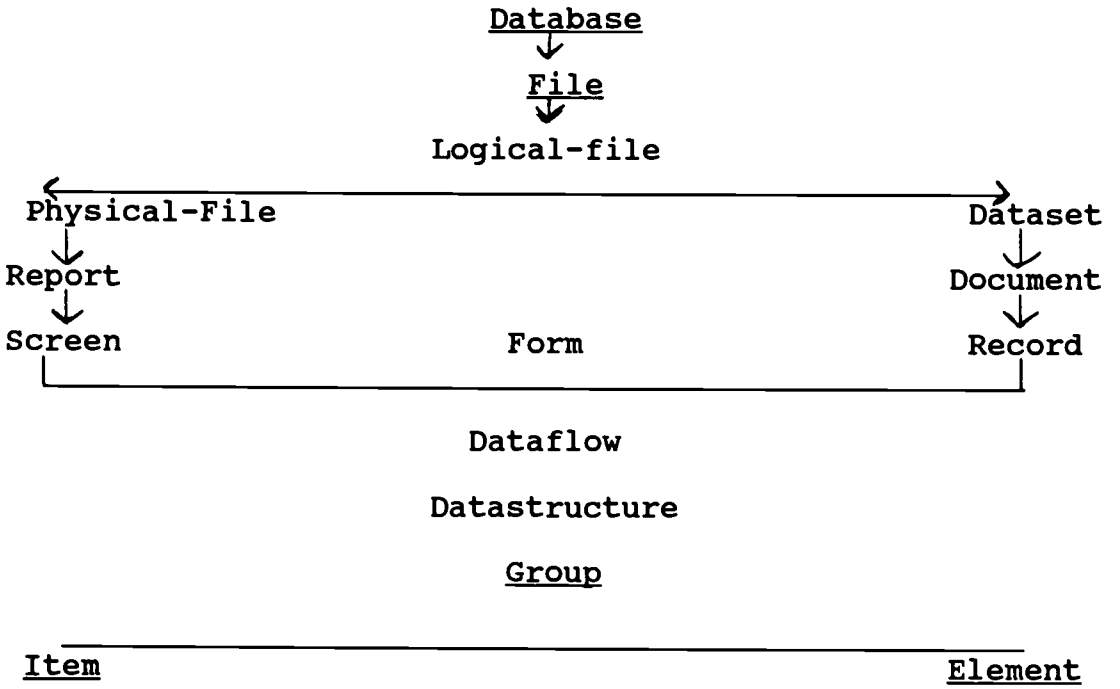


FIGURE E-1
DATAMANAGER Structured Development Structure (SDS)

Process Member Types. Process members define the structure of applications to DATAMANAGER, including the components of the application down to the program modules and subroutines. The three basic process members types are system, program, and module. The basic process members underlined in the diagram below, and the process members are grouped according to their relationship to the basic process members. Process members are defined using the syntax for the basic member type with which they are affiliated. Each member can contain other members of its type, or any higher type (e.g. an EXTERNAL-PROCESS may contain a SUBPROCESS or APPLICATION, but not a PROCESS or USER. SUBPROCESSes and APPLICATIONs are lower in the structure than the EXTERNAL-PROCESS, and therefore can be contained in an EXTERNAL-PROCESS. A PROCESS is at the same level as an EXTERNAL-PROCESS, and a USER is at a higher level, so neither may be contained by an EXTERNAL-PROCESS). The following process member hierarchy will be used.



Data Member Types. Data member types allow the identification of data elements, including their logical and physical characteristics and groupings. They can be used to describe data files, screens, reports, forms, and the actual data elements themselves. Data member types allow the standardization of data elements (e.g. a single format for dates rather than the variety of date formats currently in use). The basic data member types are file, group and item. Each basic data member type consists of a number of different data member types. The basic data members are underlined, and members are grouped according to their relationship to the basic data members.



Note that the ELEMENT member is not used to describe data, although it exists in UDS Structured Development Structure.

Supporting Member Types

General. There are two additional member types which are used to utilize a DATAMANAGER dictionary.

Command-Stream Member Type. Command-Stream members are used to facilitate functions performed by the DATAMANAGER user, such as report generation and dictionary maintenance. They contain commands which facilitate such actions as global changes, renaming of data elements, ensuring that particular relationships are maintained within the dictionary, and identification and deletion of improperly defined members.

Glossary Member Type. In order to meet the limitations on the length of member names, and to prevent member names from becoming too long to use easily, the words which make up member names must be abbreviated. There are many different sources of abbreviations, and many of them are in conflict with others. The most effective way to ensure that standard abbreviations for commonly used words are utilized uniformly within a dictionary is to maintain a list of those standard abbreviations within the dictionary. The DBC of each dictionary will utilize glossary members to maintain this abbreviation list. The UDS Structured Development Structure member type ELEMENT will be used as a glossary member.

HINTS ON AVOIDANCE OF REDUNDANCY

These are the definitions that describe the real, physical data that forms the heart of the corporate data resource. Redundancy of these dictionary entities must be controlled, reduced and minimized in order to properly manage our information resources. 'Don't reinvent the wheel' is an appropriate analogy for dictionary definitions in this category. To illustrate this point, consider the definition for the entity that is a person's last name. Generically speaking, there is only one possible thing that is a last name. There are, of course, classes of last names. There are father's last names, mother's last names, service member's last names and so forth. The data dictionary concept requires that the specific attributes for the generic entity that is a 'last name' be defined only once. The same is true for reporting unit codes, dates, monitor command codes, etc. The following examples translate this concept in DATAMANAGER terms (the specific requirements for member definitions are in the RFDD user's guide. These examples only show how a single generic definition can be used to 'drive' all other variations.)

ITEMS

(1) The skeletal definition below expresses how the generic entity 'last name' would be defined:

```
LAST-NAME
Item
Held-as Alphanumeric Null 20
```

(2) The definition below shows a variation of last name and how it is based on the above generic definition:

```
FATHERS-LAST-NAME
Item
Held-as Name LAST-NAME
```

(3) 'Name' in the above definition is a DATAMANAGER keyword.In this case it is used to tell DATAMANAGER that the member called FATHERS-LAST-NAME then, becomes a synonym for LAST-NAME. If the length attribute for last names ever changes, for example, from 20 characters to 25 characters, it need only be changed in one place: The definition for last name. This change will be automatically carried over into any definition based on the member called LAST-NAME.

GROUPS

(1) An alternative to minimizing redundancy is the use of group definitions. The below skeletal definition shows another, less desirable approach.

```
FATHERS-LAST-NAME
Group
Contains LAST-NAME
```

This approach achieves the same result as far as attributes are concerned. However, great care should be exercised because any source language generated from this group definition will show LAST-NAME as an item being contained within the group FATHERS-LAST-NAME. Using the approach for items (above), LAST-NAME will be totally transparent. Thus, definitions should be driven, so to speak, from the item level wherever possible.

Another aspect of minimizing redundancy is to avoid member names that refer to specific systems or functions. For example, FATHERS-LAST-NAME is a fathers last name, and not a JUMPS-FATHER-LAST-NAME or RPPS-FATHERS-LAST-NAME. This topic is also addressed Appendix A.

Appendix F

DATAMANAGER WORKSHEETS

SYSTEM PROCESS MEMBER WORK SHEET

Date: _____ By: _____

Approved by Controller: Date: _____ Signature: _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY *)

```
*ADD _____;
*Select 1 (ORGANIZATION, FUNCTION, USER, EXTERNAL-PROCESS,
          PROCESS, SUBPROCESS, APPLICATION, SYSTEM

*;

*MODIFY _____

*10000      DESCRIPTION                                          "
*10010      "Long title: _____"
*10020      " _____"
*10030      " _____"
*10040      " _____"
 10050      " _____"
 10060      " _____"
 10070      " _____"
 10080      " _____"
 10090      " _____"
 10100      " _____"

 12000      NOTE
 12010      "1.  Date: _____  Initials: _____"
 12020      " _____"
 12030      " _____"
 12040      " _____"

 16000      ADMINISTRATIVE-DATA
 16010      " _____"
 16020      " _____"

 18500      CATALOG
 18510      ," _____"
 18520      ," _____"

 19200      EFFECTIVE-DATA  _____

 30000       CONTAINS
 30010     ,   _____
 30020         ,_____
 30030         ,_____
 30040         ,_____
 30050         ,_____
 30060         ,_____
```

SYSTEM PROCESS MEMBER WORK SHEET (Continued)

```
32000     INPUTS
32010       _____________________
32020     , _____________________
32030     , _____________________
32040     , _____________________
32050     , _____________________
32060     , _____________________

33000     OUTPUTS
33010       _____________________
33020     , _____________________
33030     , _____________________
33040     , _____________________
33050     , _____________________
33060     , _____________________

;
```

Entered by: _____ Date: _____

PROGRAM PROCESS MEMBER WORK SHEET

Date: _____ By: _____

Approved by Controller: Date: _____ Signature: _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY *)

```
*ADD _____;
*Select 1 (JOB, PROCEDURE, TRANSACTION, JOB-STEP, or PROGRAM)
*;

*MODIFY  _____;

*10000   DESCRIPTION                                              "
*10010   "Long title: _____"
*10020   " _____"
*10030   " _____"
*10040   " _____"
 10050   " _____"
 10060   " _____"
 10070   " _____"
 10080   " _____"
 10090   " _____"
 10100   " _____"

 12000   NOTE
 12010   "1.  Date: _____  Initials: _____"
 12020   " _____"
 12030   " _____"
 12040   " _____"

 16000   ADMINISTRATIVE-DATA
 16010   " _____"
 16020   " _____"

 18500   CATALOG
 18510                    ," _____"

 19200   EFFECTIVE-DATE  _____

 30000   CONTAINS
 30010   _____
 30020   , _____
 30030   , _____
 30040   , _____
 30050   , _____
 30060   , _____
```

PROGRAM PROCESS MEMBER WORK SHEET (Continued)

```
32000     INPUTS
32010       _____________________________
32020     , _____________________________
32030     , _____________________________
32040     , _____________________________
32050     , _____________________________
32060     , _____________________________

33000     OUTPUTS
33010       _____________________________
33020     , _____________________________
33030     , _____________________________
33040     , _____________________________
33050     , _____________________________
33060     , _____________________________

34000     UPDATES
34010       _____________________________
34020     , _____________________________
34030     , _____________________________

35000     CALLS
35010       _____________________________
35020     , _____________________________
35030     , _____________________________

;
```

Entered by: _____ Date: _____

FILE ADABAS DATA MEMBER WORK SHEET
(FOR FILE ADABAS ONLY)

Date: _____ By: _____

Approved by Controller: Date: _____ Signature: _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY*)

```
*  ADD  _____ ;
*FILE ADABAS
*;

* MODIFY _____;

 20000  DEFAULTED-AS _____

 03000  ENTERED-AS 1 _____
 03100  ENTERED-AS 2 _____
 03200  ENTERED-AS 3 _____
 03300  ENTERED-AS 4 _____
 03400  ENTERED-AS 5 _____

 04000  HELD-AS 1 _____
 04100  HELD-AS 2 _____
 04200  HELD-AS 3 _____
 04300  HELD-AS 4 _____
 04400  HELD-AS 5 _____

 05000  REPORTED-AS 1 _____
 05100  REPORTED-AS 2 _____
 05200  REPORTED-AS 3 _____
 05300  REPORTED-AS 4 _____

*10000  DESCRIPTION
*10010  "LONG TITLE: _____
*10020  " _____"
*10030  " _____"
*10040  " _____"
*10050  " _____"
 10060  " _____"
 10070  " _____"
 10080  " _____"
 10090  " _____"
 10100  " _____"
 10110  " _____"
 10120  " _____"

 12000  NOTE
 12010  "1.  DATE: _____  INITIALS: _____"
 12020  " _____"
 12030  " _____"
 12040  " _____"
```

FILE ADABAS-DATABASE MEMBER WORK SHEET (Continued)

```
16000   ADMINISTRATIVE-DATA
16010   " _____"
16020   " _____"

18000   ALIAS '
18070       "REF _____"
18080       "INT _____"
18090      ,"MAINT _____"
18100      ,DEN " _____"
18110      ,COBOL " _____"

18500   CATALOG
18510       " _____"
18520        ,"FILENR= _____"
18530        ,"ADABAS _____"
18540        ," _____"
18550        ," _____"

19200   EFFECTIVE-DATE _____

19700   SEE _____
19710         , _____

20000   FIELD-NAMES
20010   _____ is _____
20020   _____ is _____
20030   _____ is _____

26000   DESCRIPTORS
26010   _____ (UNIQUE OR BLANK) _____"
26020   , _____ (UNIQUE OR BLANK) _____"
26030   , _____ (UNIQUE OR BLANK) _____

28000   SUPER-DESCRIPTORS                    (Bytes Optional)
28010   _____ is _____ BYTES _____ TO _____
28020   _____ is _____ BYTES _____ TO _____
28030   _____ is _____ BYTES _____ TO _____
28040   WITH _____ BYTES _____ TO _____

29000   SUB-DESCRIPTORS
29010   _____ is _____ BYTES _____ TO _____
29020   , _____ is _____ BYTES _____ TO _____

29500   PHONETIC-NAMES
29510   _____ is _____
29520   , _____ is _____
```

FILE ADABAS DATA MEMBER WORK SHEET (Continued)

```
30000    CONTAINS
30010         , _____
30020         , _____
30030         , _____
30040         , _____
30050         , _____
30060         , _____
```

Entered By: _____ Date: _____

MODULE PROCESS MEMBER WORK SHEET

Date: _____ By: _____

Approved by Controller: Date: _____ Signature: _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY *)

```
* ADD _____;
*Select  1  (MODULE, SUBROUTINE, or DATASTORE)

*MODIFY  _____;
                                                '
*10000      DESCRIPTION
*10010      "Long title: _____"
*10020      " _____"
*10030      " _____"
*10040      " _____"
 10050      " _____"
 10060      " _____"
 10070      " _____"
 10080      " _____"
 10090      " _____"
 10100      " _____"

 10200      NOTE
 12010      "1.  Date: _____ Initials: _____"
 12020      " _____"
 12030      " _____"
 12040      " _____"

 16000      ADMINISTRATIVE-DATA
 16010      " _____"
 16020      " _____"

 18500      CATALOG
 18510                ,"_____"

 19200      EFFECTIVE-DATE _____

 30000      CONTAINS
 30010       _____
 30020         , _____
 30030         , _____
 30040         , _____
 30050         , _____
 30060         , _____
```

PROGRAM PROCESS MEMBER WORK SHEET (Continued)

32000 INPUTS
32010 _____
32020 , _____
32030 , _____
32040 , _____
32050 , _____
32060 , _____

33000 OUTPUTS
33010 _____
33020 , _____
33030 , _____
33040 , _____
33050 , _____
33060 , _____

34000 UPDATES
34010 _____
34020 , _____
34030 , _____

35000 CALLS
35010 _____
35020 , _____
35030 , _____

Entered by: _____ Date: _____

FILE DATA MEMBER WORK SHEET

For FILE, LOGICAL-FILE, PHYSICAL-FILE, DATASET, REPORT, DOCUMENT

Date: _____ By: _____

Approved by Controller: Date: _____ Signature: _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY *)

```
*ADD _____;
*Choose One (FILE, LOGICAL-FILE, PHYSICAL-FILE, DATASET, REPORT,
             DOCUMENT)

*;

*MODIFY _____;

*10000      DESCRIPTION
*10010      "Long title: _____"
*10020      " _____"
*10030      " _____"
*10040      " _____"
 10050      " _____"
 10060      " _____"
 10070      " _____"
 10080      " _____"
 10090      " _____"
 10100      " _____"

 12000      NOTE
 12010      "1.  Date: _____  Initials: _____"
 12020      " _____"
 12030      " _____"
 12040      " _____"

 16000      ADMINISTRATIVE-DATA
 16010      " _____"
 16020      " _____"

 18000      ALIAS
 18070           "REF _____"
 18080           "INIT _____"
 18090           ,"MAINT _____"
 18100           ,"DEN  " _____"
 18160           ,"COBOL " _____"
```

FILE DATA MEMBER WORK SHEET (Continued)

```
18500     CATALOG "IDF"
18510           ,"SOURCE= _____________________________________"
18520           ," _____________________________________"
18530           ," _____________________________________"
18540           ," _____________________________________"
18550           ," _____________________________________"

19220     EFFECTIVE-DATE _____________________________________

19700     SEE _____________________________________
19710       , _____________________________________

40000     ENTERED-AS 1 _____________________
40100     ENTERED-AS 2 _____________________
40200     ENTERED-AS 3 _____________________
40300     ENTERED-AS 4 _____________________
40400     ENTERED-AS 5 _____________________

41000     HELD-AS 1 _____________________
41100     HELD-AS 2 _____________________
41200     HELD-AS 3 _____________________
41300     HELD-AS 4 _____________________
41400     HELD-AS 5 _____________________

42000     REPORTED-AS 1 _____________________
42100     REPORTED-AS 2 _____________________
42200     REPORTED-AS 3 _____________________
42300     REPORTED-AS 4 _____________________

43000     DEFAULTED-AS _____________________

44000     CONTAINS
44010       _____________________________________
44020     , _____________________________________
44030     , _____________________________________
44040     , _____________________________________
44050     , _____________________________________
44060     , _____________________________________

45000     Select 1 (SEQUENTIAL, INDEXED, DIRECT, KEYED, VSAM
                    PARTITIONED)

45500     Select 1 (FIXED, VARIABLE, UNDEFINED, SPANNED)

46000     SORT-KEY _____________________

46200     GENERATION-CYCLE _____________________________________
```

Entered by: _____ Date: _____

GROUP WORK SHEET

Used for SCREEN, FORM, RECORD, DATAFLOW, DATASTRUCTURE and GROUP

Date: _____ By: _____

Approved by Controller: Date: _____ _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY*)

```
* ADD _____;
*Circle Proper Member Type (SCREEN, FORM, RECORD, DATAFLOW,
                            DATASTRUCTURE of GROUP)

*;

* MODIFY _____;

 02000    DEFAULTED-AS    _____

 03000    ENTERED-AS 1    _____
 03100    ENTERED-AS 2    _____
 03200    ENTERED-AS 3    _____
 03300    ENTERED-AS 4    _____
 03400    ENTERED-AS 5    _____

 04000    HELD-AS 1       _____
 04100    HELD-AS 2       _____
 04200    HELD-AS 3       _____
 04300    HELD-AS 4       _____
 04400    HELD-AS 5       _____

 05000    REPORTED-AS 1   _____
 05100    REPORTED-AS 2   _____
 05200    REPORTED-AS 3   _____
 05300    REPORTED-AS 4   _____
 05400    REPORTED-AS 5   _____

*10000    DESCRIPTION
*10010    "Long title: _____"
*10020    " _____"
*10030    " _____"
*10040    " _____"
*10050    " _____"
 10060    " _____"
 10070    " _____"
 10080    " _____"
 10090    " _____"

 12000    NOTE
 12010    "1.  Date: _____  Initials: _____"
 12020    " _____"
 12030    " _____"
 12040    " _____"
```

GROUP WORK SHEET (Continued)

```
16000     ADMINISTRATIVE-DATA
16010      " _____"
16020      " _____"

18000     ALIAS
18070          REF    " _____"
18080         ,INIT   " _____"
18090         ,MAINT  " _____"
18100         ,DEN    " _____"
18160         ,COBOL  " _____"

18500     CATALOG "IDF"
18510            ,"SOURCE= _____"
18520            ," _____"
18530            ," _____"
18540            ," _____"
18550            ," _____"

19200     EFFECTIVE-DATA  _____

19700     SEE _____
19710       , _____

50000     CONTAINS
50010           _____
50020       , _____
50030       , _____
50040       , _____
50050       , _____
50060       , _____

;
```

Entered by: _____ Date: _____

ITEM WORK SHEET
(For ITEM only)

Date: _____ By: _____

Approved by Controller: Date: _____ Signature: _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY *)

```
* ADD _____;
*ITEM
*;

* MODIFY _____;
*02000    DEFAULTED-AS
*02100       , (for ADABAS enter Variable, null, fixed or blank)-
*02200       , _____"

03000     ENTERED-AS 1 _____ " _____" _____"
03100     ENTERED-AS 2 _____ " _____" _____"
03200     ENTERED-AS 3 _____ " _____" _____"
03300     ENTERED-AS 4 _____ " _____" _____"
03400     ENTERED-AS 5 _____ " _____" _____"

04000     HELD-AS 1 _____ " _____" _____"
04100     HELD-AS 2 _____ " _____" _____"
04200     HELD-AS 3 _____ " _____" _____"
04300     HELD-AS 4 _____ " _____" _____"
04400     HELD-AS 5 _____ " _____" _____"

05000     REPORTED-AS 1 _____ " _____" _____"
05100     REPORTED-AS 2 _____ " _____" _____"
05200     REPORTED-AS 3 _____ " _____" _____"
05300     REPORTED-AS 4 _____ " _____" _____"
05400     REPORTED-AS 5 _____ " _____" _____"

*10000    DESCRIPTION
*10010       "Long title: _____"
*10010       " _____"
*10020       " _____"
*10030       " _____"
*10040       " _____"
 10050       " _____"
 10060       " _____"
 10070       " _____"
 10080       " _____"
 10090       " _____"
 10100         _____"

12000     NOTE
12010        "1.  Date: _____  Initials: _____"
12020        " _____"
12030        " _____"
12040        " _____"
```

ITEM WORK SHEET (Continued)

```
16000     ADMINISTRATIVE-DATA
16010          " _____________________________________________"
16020          " _____________________________________________"

18000     ALIAS
18070           REF     " _____________________"
18080          ,INIT    " _____________________"
18090          ,MAINT   " _____________________"
18100          ,DEN     " _____________________"
18160          ,COBOL   " _____________________"

*18500    CATALOG   "IDF"
*18510              ,"SOURCE= _____________________"
 18520              ," _____________________"
 18530              ," _____________________"
 18540              ," _____________________"
 18550              ," _____________________"
 18560              ," _____________________"
 18570              ," _____________________"
 18580              ," _____________________"
 18590              ," _____________________"
 18600              ," _____________________"
 18610              ," _____________________"
*18620              ," Source= _____________"

 19000    ACCESS-AUTHORITY
*19010    , _____________________________________"
*19020    , _____________________________________"
*19030    , _____________________________________"
*19040    , _____________________________________"
*19050    , _____________________________________"

 19200    EFFECTIVE-DATE _____________________

;
```

Entered by: _____ Date: _____

DFD PROCESS WORK SHEET

Date: _____ By: _____

Approved by Controller: Date: _____ Signature: _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY *)

```
* ADD _____;
*SYSTEM
*;

* MODIFY _____;

*10000    DESCRIPTION
*10010    "Long title: _____"
*10020    " _____"
*10030    " _____"
*10040    " _____"
*10050    " _____"
*10060    " _____"
*10070    " _____"
 10080    " _____"
 10090    " _____"
 10100    " _____"

 12000    NOTE
 12010    "1.  Date: _____  Initials: _____"
 12020    " _____"
 12030    " _____"
 12040    " _____"

 16000    ADMINISTRATIVE-DATA
 16010    " _____"
 16020    " _____"

*18000    ALIAS
*18060    LEVEL " _____"

*18500    CATALOG "DFD"
 18510           ," _____"

 19200    EFFECTIVE-DATE _____

*30000    CONTAINS
*30010     _____
 30020        , _____
 30030        , _____
 30040        , _____
 30050        , _____
 30060        , _____
```

DFD PROCESS WORK SHEET

32000 INPUTS
32010 _____
32020 , _____
32030 , _____
32040 , _____
32050 , _____
32060 , _____

33000 OUTPUTS
33010 _____
33020 , _____
33030 , _____
33040 , _____
33050 , _____
33060 , _____

;

Entered by: _____ Date: _____

DFD SYSTEM WORK SHEET

Date: _____ By: _____

Approved by Controller: Date: _____ Signature: _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY *)

```
* ADD _____;
*SYSTEM
*;

* MODIFY _____;

*10000    DESCRIPTION
*10010    "Long title: _____"
*10020    " _____"
*10030    " _____"
*10040    " _____"
*10050    " _____"
*10060    " _____"
*10070    " _____"
 10080    " _____"
 10090    " _____"
 10100    " _____"

 12000    NOTE
 12010    "1.  Date: _____  Initials: _____"
 12020    " _____"
 12030    " _____"
 12040    " _____"

 16000    ADMINISTRATIVE-DATA
 16010    " _____"
 16020    " _____"

*18000    ALIAS
*18060    LEVEL " _____"

*18500    CATALOG "DFD"
 18510           ," _____"

 19200    EFFECTIVE-DATE _____"

 30000    CONTAINS
 30010           _____
 30020         , _____
 30030         , _____
 30040         , _____
 30050         , _____
 30060         , _____
```

Entered by: _____ Date: _____

DFD DATAFLOW WORK SHEET

Date: _____ By: _____

Approved by Controller: Date: _____ Signature: _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY *)

```
*  ADD  _____;
* SYSTEM
*;

*  MODIFY  _____;

*10000     DESCRIPTION
*10010     "Long title;  _____"
*10030     "_____"
*10040     "_____"
*10050     "_____"
*10060     "_____"
*10070     "_____"
 10080     "_____"
 10090     "_____"
 10100     "_____"

 12000     NOTE
 12010     "1.  Date:  _____  Initials:  _____
 12020     "_____"
 12030     "_____"
 12040     "_____"

 16000     ADMINISTRATIVE-DATA
 16010     "_____"
 16020     "_____"

*18000     ALIAS
*18060     LEVEL  "_____"

*18500     CATALOG "DFD"
 18510           ,"_____"

 19200     EFFECTIVE-DATE  _____

 30000     CONTAINS
 30010           _____
 30020         , _____
 30030         , _____
 30040         , _____
 30050         , _____
 30060         , _____
```

Entered by: _____ Date: _____

DFD EXTERNAL-PROCESS WORK SHEET

Date: _____ By: _____

Approved by Controller: Date: _____ Signature: _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY *)

```
*  ADD _____;
* EXTERNAL PROCESS
*;

*  MODIFY _____;

*10000    DESCRIPTION
*10010    "Long title; _____"
*10030    "_____"
*10040    "_____"
 10050    "_____"
 10060    "_____"
 10070    "_____"
 10080    "_____"
 10090    "_____"
 10100    "_____"

 12000    NOTE
 12010    "1.  Date: _____ Initials: _____
 12020    "_____"
 12030    "_____"
 12040    "_____"

 16000    ADMINISTRATIVE-DATA
 16010    "_____"
 16020    "_____"

*18000    ALIAS
*18060    LEVEL  "_____"

*18500    CATALOG "DFD"
 18510           ,"_____"

 19200    EFFECTIVE-DATE  _____

 30000    CONTAINS
 30010             _____
 30020          , _____
 30030          , _____
 30040          , _____
 30050          , _____
 30060          , _____
```

DFD EXTERNAL PROCESS WORK SHEET (Continued)

```
32000     INPUTS
32010       _____________________________
32020     , _____________________________
32030     , _____________________________
32040     , _____________________________
32050     , _____________________________
32060     , _____________________________

33000     OUTPUTS
33010       _____________________________
33020     , _____________________________
33030     , _____________________________
33040     , _____________________________
33050     , _____________________________
33060     , _____________________________

;
```

Entered by: _____ Date: _____

DFD DATASTORE WORK SHEET

Date: _____ By: _____

Approved by Controller: Date: _____ Signature: _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY *)

```
*  ADD _____;
* DATASTORE
*;

*  MODIFY _____;

*10000    DESCRIPTION
*10010    "Long title; _____"
*10020    "_____"
*10030    "_____"
 10040    "_____"
 10050    "_____"
 10060    "_____"
 10070    "_____"
 10080    "_____"
 10090    "_____"
 10100    "_____"

 12000    NOTE
 12010    "1.  Date: _____ Initials: _____
 12020    "_____"
 12030    "_____"
 12040    "_____"

 16000    ADMINISTRATIVE-DATA
 16010    "_____"
 16020    "_____"

*18000    ALIAS
*18060    LEVEL  "_____"

*18500    CATALOG "DFD"
 18510           ,"_____"

 19200    EFFECTIVE-DATE  _____

 30000    CONTAINS
 30010          _____
 30020         , _____
 30030         , _____
 30040         , _____
 30050         , _____
 30060         , _____
```

DFD DATASTORE WORK SHEET (Continued)

```
32000    INPUTS
32010      _____________________________
32020    , _____________________________
32030    , _____________________________
32040    , _____________________________
32050    , _____________________________
32060    , _____________________________

33000    OUTPUTS
33010      _____________________________
33020    , _____________________________
33030    , _____________________________
33040    , _____________________________
33050    , _____________________________
33060    , _____________________________

;
```

Entered by: _____ Date: _____

COMMAND-STREAM WORK SHEET
(For COMMAND-STREAM only)

Date: _____ By: _____

Approved by Controller: Date: _____ Signature: _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY *)

```
*  ADD _____;
* COMMAND-STREAM
*;

*  MODIFY _____;

*10000     DESCRIPTION
*10010     "Long title; _____"
*10020     "_____"
*10030     "_____"
*10040     "_____"
*10050     "_____"
*10060     "_____"
*10070     "_____"

 19200     EFFECTIVE-DATE _____

 19700     SEE _____

*20000     CONTENTS
 20010           _____
 20020           _____
 20030           _____
*29990         ; _____

;
```

Entered by: _____ Date: _____

GLOSSARY WORK SHEET
(For ELEMENT only)

Date: _____ By: _____

Approved by Controller: Date: _____ Signature: _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY *)

```
*  ADD  _____;
* ELEMENT
*;

*  MODIFY  _____;

*10000     DESCRIPTION
*10010     "Long title;  _____"
*10020     "_____"
*10030     "_____"
*10040     "_____"
*10050     "_____"
*10060     "_____"
*10070     "_____"

*18000     ALIAS
*18010         "FULL-SPELLING=full spelling"

*18500     CATALOG     "GLOSSARY"

 19200     EFFECTIVE-DATE  _____

 19700     SEE  _____

;
```

Entered by: _____ Date: _____

FILE ADABAS-DATABASE MEMBER WORK SHEET
(FOR ADABAS-DATABASE ONLY)

Date: _____ By: _____

Approved by Controller: Date: _____ Signature: _____

ITEMS MARKED WITH * MUST BE ENTERED (DO NOT KEY *)

```
*  ADD or REPLACE (circle one) ____________________________________

* ADABAS-DATABASE
*;
*  MODIFY ____________________________________________;

*10000     DESCRIPTION
*10010     "Long title; ____________________________________________"
*10020     "____________________________________________________________"
*10030     "____________________________________________________________"
*10040     "____________________________________________________________"
 10050     "____________________________________________________________"
 10060     "____________________________________________________________"
 10070     "____________________________________________________________"
 10080     "____________________________________________________________"
 10090     "____________________________________________________________"
 10100     "____________________________________________________________"

 12000     NOTE
 12010     "1.  Date: _____ Initials: _____"
 12020     "____________________________________________________________"
 12030     "____________________________________________________________"
 12040     "____________________________________________________________"

 16000     ADMINISTRATIVE-DATA
 16010     "____________________________________________________________"
 16020     "____________________________________________________________"

 18000     ALIAS
 18010     "_____________________________"
 18020    ,"_____________________________"
 18030    ,"_____________________________"
 18040    ,"_____________________________"
 18050    ,"_____________________________"
 18060    ,"_____________________________"
 18070    ,"_____________________________"
 18080    ,"_____________________________"
 18090    ,"_____________________________"
 18100    ,"_____________________________"
 18110    ,"_____________________________"
```

FILE ADABAS-DATABASE MEMBER WORK SHEET (Continued)

```
 18500    CATALOG
 18510    "TYPE=ADABAS _____"
 18520    ,"_____"
 18530    ,"_____"
 18540    ,"_____"
 18550    ,"_____"

 18570    ,"_____"
 18580    ,"_____"
 18590    ,"_____"
 18600    ,"_____"
 18610    ,"_____"
 18620    ,"_____"

*19000    ACCESS-AUTHORITY
*19010     _____
*19020   ,_____
*19030   ,_____
*19040   ,_____

 19200    EFFECTIVE-DATE _____

 30000    CONTAINS
 30010     _____
 30020   ,_____
 30030   ,_____
 30040   ,_____
 30050   ,_____
 30060   ,_____

 39400    COUPLE _____

 39600    CIPHER _____

 46800    DEVICE _____
```

Entered by: _____ Date: _____

Appendix G

DATAMANAGER LINE NUMBERING CONVENTIONS

1. Common Clauses

Line Range

10000-11990	DESCRIPTION	200
10000	"Long Title: "	
10010	"_____"	
12000-13990	NOTE	200
	Each note will be prefaced with:	
	n. Date. YYMMDD. Initials: FML.	
14000-15990	COMMENTS	200
	Each comments will be prefaced with:	
	N. Date. YYMMDD. Initials: FML.	
16000-17990	ADMINISTRATIVE-DATA	200
	Each entry will be prefaced with:	
	n. Date. YYMMDD. Initials: FML.	
18000-18490		50
18000	ALIAS	
18010		
18020		
18030		
18040		
18050		
18060		
18070	REF	
18080	INIT	
18090	MAINT	
18100	DEN	
18110	ADABAS	
18120	IMS	
18130	TOTAL	
18140	PL/1	
18150	ASSEMBLER	
18160	COBOL	
18500-18990	CATALOGUE	50
19000-19190	ACCESS-AUTHORITY	20
19200-19290	EFFECTIVE-DATE	20
19300-19390	FREQUENCY	10
19400-19490	OBSOLETE-DATE	10
19500-19590	QUERY	10
19600-19690	SECURITY-CLASSIFICATION	10
19700-19790	SEE	10

DATAMANAGER LINE NUMBERING CONVENTION (Continued)

2. COMMAND-STREAM MEMBERS

20000-29990	CONTENTS	

3. Process Members.

30000-31990	CONTAINS	200
32000-32990	INPUTS	100
33000-33990	OUTPUTS	100
34000-34990	UPDATES	100
35000-35990	CALLS	100
36000-36990	PARAMETERS	100
37000-37190	LANGUAGE	20
37200-37390	AUTHOR	20
37400-37590	INSTALLATION	20
37600-37790	DATE-WRITTEN	20
37800-37990	SOURCE-COMPUTER	20
38000-38190	OBJECT-COMPUTER	20
38200-38390	SPECIAL-NAMES	20
38400-38590	I-O-CONTROL	20
38600-38790	ASSIGNMENT	20
38800-38990	EDIT-INPUT	20
39000-39190	EDIT-OUTPUT	20
39200-39390	EDIT-OUTDATE	20

4. FILE Members.

40000-40990	ENTERED-AS	100
41000-41990	HELD-AS	100
42000-42990	REPORTED-AS	100
43000-43990	DEFAUL-AS	100
20000-25990	FIELD-NAMES	
26000-27990	DESCRIPTORS	
28000-28990	SUPER-DESCRIPTORS	
29000-29490	SUB-DESCRIPTORS	
29500-29990	PHONETIC-NAMES	
44000-44990	CONTAINS	100
45000-45490	SEQUENTIAL	50
	INDEXED	
	DIRECT	
	KEYED	
	VSAM	
	PARTITIONED	

DATAMANAGER LINE NUMBERING CONVENTION (Continued)

45500-45990	FIXED	50
	VARIABLE	
	UNDEFINED	
	SPANNED	
46000-46190	SORT-KEY	20
46200-46390	GENERATION-CYCLE	20
46400-46590	RETENTION-PERIOD	20
46600-46790	GROWTH-RATE	20
46800-46990	DEVICE	20
47000-47190	VOLUME	20
47200-47390	SIZE	20
47400-47590	LABELS	

5. GROUP Member.

02000	DEFAULTED-AS	
03000	ENTERED-AS 1	
03100	ENTERED-AS 2	
03200	ENTERED-AS 3	
03300	ENTERED-AS 4	
03400	ENTERED-AS 5	
04000	HELD-AS 1	
04100	HELD-AS 2	
04200	HELD-AS 3	
04300	HELD-AS 4	
04400	HELD-AS 5	
05000	REPORTED-AS 1	
05100	REPORTED-AS 2	
05200	REPORTED-AS 3	
05300	REPORTED-AS 4	
05400	REPORTED-AS 5	
50000-50990	CONTAINS	100
51000-51990	KEYS	100
52000-52990	USER-EXIT	100

6. DATA DESCRIPTION CLAUSES

03000-03990	ENTERED-AS 1	50
	ENTERED-AS 2	50
	ENTERED-AS 3	50
	ENTERED-AS 4	50
	ENTERED-AS 5	50
	ENTERED-AS 6	50

DATAMANAGER LINE NUMBERING CONVENTION (Continued)

	ENTERED-AS 7	50
	ENTERED-AS 8	50
	ENTERED-AS 9	50
	ENTERED-AS 10	50
	ENTERED-AS 11	50
	ENTERED-AS 12	50
	ENTERED-AS 13	50
	ENTERED-AS 14	50
	ENTERED-AS 15	50
04000-04990	HELD-AS 1	50
	HELD-AS 2	50
	HELD-AS 3	50
	HELD-AS 4	50
	HELD-AS 5	50
	HELD-AS 6	50
	HELD-AS 7	50
	HELD-AS 8	50
	HELD-AS 9	50
	HELD-AS 10	50
	HELD-AS 11	50
	HELD-AS 12	50
	HELD-AS 13	50
	HELD-AS 14	50
	HELD-AS 15	50
05000-05990	REPORTED-AS 1	50
	REPORTED-AS 2	50
	REPORTED-AS 3	50
	REPORTED-AS 4	50
	REPORTED-AS 5	50
	REPORTED-AS 6	50
	REPORTED-AS 7	50
	REPORTED-AS 8	50
	REPORTED-AS 9	50
	REPORTED-AS 10	50
	REPORTED-AS 11	50
	REPORTED-AS 13	50
	REPORTED-AS 14	50
	REPORTED-AS 15	50
90000-90490	DEFAULTED-AS	50

Appendix H

USE OF DATAMANAGER FOR DATA FLOW DIAGRAMS

GENERAL.

The process of manually constructing and maintaining DFD's is difficult, especially when attempting to make all levels consistent. DATAMANAGER provides an automated tool to facilitate creation and maintenance of DFD's.

COMPONENTS OF DATA FLOW DIAGRAMS.

Data Flow Diagrams consist of four components: data flows, processes, data stores, and data originator/terminators. These four components are used to describe an existing or proposed system.

USING DATAMANAGER TO REPRESENT DATA FLOW DIAGRAMS

SYSTEM Systems being described using data flow diagrams are represented in DATAMANAGER by the FUNCTION member type. The DFD FUNCTION WORK SHEET will be used to gather information on the DFD data flows.

At a minimum the following information is required:

10000 18500	DESCRIPTION	As per Work Sheet "DFD" should be entered, along with the name of the system being diagrammed.

If the system being diagrammed is composed of DFD processes and DFD data originator/terminators, this relationship is indicated by use of the CONTAINS clause.

DATA FLOWS

DFD data flows are represented in DATAMANAGER by the DATAFLOW member type. The DFD DATAFLOW WORK SHEET will be used to gather information on the DFD data flows.

At a minimum the following information is required:

10000	DESCRIPTION	As per Work Sheet
18000	ALIAS	
18060	LEVEL "x"	Where x is the appropriate Level number "DFD" should
18500		be entered, along with the name of the system being diagrammed.

If the DFD data flow can be decomposed into other data flows (or specific documents or reports), then those component items will be indicated in the CONTAINS clause.

Processes.

DFD processes are represented in DATAMANAGER by the PROCESS member type. The DFD PROCESS WORK SHEET will be used to gather information on the DFD processes.

At a minimum the following information is required:

10000	DESCRIPTION	As per Work Sheet
18000	ALIAS	
18060	LEVEL "x"	Where x is the appropriate Level number "DFD" should be entered, along with the
18500	CATALOG	name of the system being diagrammed.

DFD processes input and output DFD data flows. This input and output is represented by the INPUT and OUTPUT clauses. Process members may only input or output data flows which are on the same level as the process member.

If the DFD process is composed of other DFD processes, then those component processes are indicated in the CONTAINS clause.

Data Stores.

DFD data stores are represented in DATAMANAGER by the DATASTORE member type. The DFD DATASTORE MEMBER WORK SHEET will be used to gather information on DFD Data stores.

At a minimum the following information is required:

10000	DESCRIPTION	As per Work Sheet
18000	ALIAS	
18060	LEVEL "x"	Where x is the appropriate Level number "DFD" should be entered,
18500	CATALOG	along with the name of the system being diagrammed.

DFD data stores input and output DFD data flows are represented by the INPUT and OUTPUT clauses. Data store members may only input or output data flows which are on the same level as the data store member. If the DFD data store is composed of other DFD Data stores, then those component processes are indicated in the CONTAINS clause.

Data Originator/Terminator

DFD data originator/terminators are represented in DATAMANAGER by the EXTERNAL-PROCESS member type. The DFD EXTERNAL-PROCESS MEMBER WORK SHEET will be used to gather information on DFD data originator/terminator.

At a minimum the following information is required:

10000	DESCRIPTION	As per Work Sheet
18000	ALIAS	
18060	LEVEL "x"	Where x is the appropriate Level number "DFD" should be entered, along with the
18500	CATALOG	name of the system being diagrammed.

DFD data originator/terminators input and output DFD Data Flows. This input and output is represented by the INPUT and OUTPUT clauses. Data originator/terminators members may only input or output data flows which are on the same level as the data originator/terminator member.

If the DFD data originator/terminator is composed of other DFD data originator/terminators, then those component data originator/terminators are indicated in the CONTAINS clause.

Reports and Documents. DFD Reports and Document will be identified in DATAMANAGER as REPORTS and DOCUMENTS.

Levels. As indicated in the preceding paragraphs, DFD levels are represented by the ALIAS LEVEL "n" attribute. The LEVEL ALIAS will contain the level as which the DFD component is being defined. The ALIAS LEVEL "n" clause can contain the values "0" through "100".

Naming DFD Members in DATAMANAGER.

The standard method of naming DFD processes in DATAMANAGER is to preface the name of the DFD process with the number of the DFD process as defined above, replacing the periods with hyphens. For instance, DFD process Create-MENS (numbered as process 1.2) which is defined as the second process at Level 1 within the first process at Level 0 would be named 1-2-CREATEJ-MENS. DFD process Concept Development, which is the first process at level 0 (numbered as process 1) will be names 1-Concept-Development. Use of this convention will ensure that all DFD processes are listed in the sequence in which they are created. Data stores, data flows, and data originator/terminators are merely names. They are not prefixed with their level/process number.

(This page intentionally left blank)

Appendix I

BIBLIOGRAPHY

The following technical documentation on DATAMANAGER is available from the vendor, Manager Software Products:

DATAMANAGER User's Guide

DATAMANAGER Messages

DATAMANAGER Automation of Set Up

DATAMANAGER Source Language Generation

DATAMANAGER Introduction to Status

DATAMANAGER User Defined Syntax

DATAMANAGER User Interface Facility

DATAMANAGER ADABAS Interface

DATAMANAGER Controller's Manual
(Note: This document is restricted in distribution as if contains proprietary vendor information. It is available through the Data Base Management and Data Base Administration Offices.)

The following references provide additional information in the use of a data dictionary.

Flavin, Matt, Fundamental Concepts of Information Modeling, New York, NY, Prentice-Hall, 1981.

Martin, James, Managing the Data Base Environment, Englewood Cliffs, NJ, Prentice-Hall, 1983.

(This page intentionally left blank)

Appendix J

GLOSSARY

ATTRIBUTE: A quality or characteristic of an entity or standard element, having a name, definition and value. The characteristics that are used to define a standard element. An item of information about an element. Properties that describe data objects. Also known as metadata.

AUTOMATED INFORMATION SYSTEM (AIS): A combination of information, computer, and telecommunications resources and other information technology that collects, records, processes, stores, communicates, retrieves, and displays data.

CLASSIFICATION: The process of breaking down a general group of data elements into specific categories.

DATA: Representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by humans or by automatic means. Any representations such as characters or analog quantities to which meaning is, or might be, assigned.

DATA ADMINISTRATION: The responsibility for definition, organization, supervision, and protection of data within an enterprise or organization.

DATA ADMINISTRATOR (DA): A person or group that ensures the utility of data used within an organization by defining data policies and standards, planning for the efficient use of data, coordinating data structures among organizational components, performing logical data base designs, and defining data security procedures.

DATABASE: A collection of interrelated data, often with controlled redundancy, organized according to a schema to serve one or more applications; the data are stored so that they can be used by different programs without concern for the data structure or organization.

DATABASE ADMINISTRATOR: A person or group that enforces policy on "how", "where", and "in what manner" data is stored and maintained in each database. Provides information to the DA on organizational use of data within the subject database.

DATA CODE: A number, letter, character, symbol or any combination thereof (a system of valid symbols) used to represent literal data. A form of qualitative data.

DATA DICTIONARY: A specialized type of database containing metadata that is managed by a data dictionary system; a repository of information describing the characteristics of data

used to design, monitor, document, protect, and control data in ISs and databases; an application of a data dictionary system.

DATA ELEMENT: A basic unit of information having a meaning and subcategories (data items) of distinct units and values.

DATA ELEMENT NAME: A unique, meaningful name applied to an information element to distinguish it from other data entities. The name provides a basis for categorization of like data elements.

DATA INTEGRITY: Embodies the concepts that the database management system (DBMS) will perform its function consistently, preserve data without unintentional change, produce results that are correct to the defined degree of precision, and maintain data availability.

DATA ITEM: A subunit of descriptive information or value classified under a data element.

DATA MODEL: Identifies the data, their attributes, and relationships or associations with other data.

DATA VALUE: A value associated with a data element. One of the allowable values of a data element. Synonym of "a data item".

DATAMANAGER: A Management Systems & Programming, Ltd. (MSP) corporate dictionary/repository driven Information Resource Management System containing information concerning data, processes and their relationships. A central control facility that provides the basis for shared and consistent system resource management. As a repository of descriptive data, DATAMANAGER is used to manage definitions and syntax of data elements, enforce data element naming standards, support data protection, control resources and create data relationships.

FUNCTIONAL AREA: A range of subject matter grouped under a single heading because of its similarity in use or genesis.

IMAGERY: Collectively, the representations of objects reproduced electronically or by optical means on file, electronic display devices, or other media.

INFORMATION: The meaning that a human assigns to data by means of the known conventions used in their representation.

INFORMATION RESOURCE DICTIONARY SYSTEM (IRDS): A set of standard specifications for a data dictionary system resulting from U.S. Federal and national standards efforts; a computer software system conforming to those standards that provides facilities for recording, storing, and processing descriptions of an organization's significant information and information processing resources.

INFORMATION SYSTEM (IS): A combination of information, information technology, and personnel resources that collects, records, processes, stores, communicates, retrieves, and displays either manually or with varying degrees of automation.

METADATA: Information describing the characteristics of data; data or information about data; and descriptive information about an organization's data, data activities, systems, and holdings.

SCHEMA: A structural (logical, as opposed to physical) description of a data base. A description of a data base in terms of the data characteristics and relationships. The representation or definition of the structure of a data base.

STANDARD: An exact value, a physical entity, or an abstract concept established and defined by authority, custom, or common consent to serve as a reference, model, or rule in measuring quantities or qualities , establishing practices or procedures, or evaluating results. A fixed quantity or quality.

STANDARD DATA ELEMENT: Data element registered IAW DoD data administration procedures.

SYMBOLOGY: Any graphic representation of concepts or physical objects.

(This page intentionally left blank)

COMMENTS/REVISIONS

Technical publications under the Information Resources Management (IRM) Standards and Guidelines Program (MCO 5271.1) are reviewed annually. Your comments and/or recommendations are strongly encouraged:

IRM Tech Pub Name: _____

IRM-____-___ (Number) Date of Tech Pub: _____

COMMENTS/RECOMMENDATIONS:_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Name/Rank:_____(optional)

Unit:_____(optional)

Mail To: Director
U.S. Marine Corps Computer and Telecom Activity
Building 2006, HQBN, MCCDC
Quantico, VA 22134-5010

ELMS - GICISN:MQGMCCTA