AFWAL-TR-86-4006
Volume V
Part 3
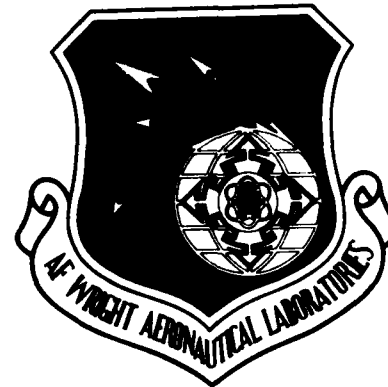
AD-A181 239

INTEGRATED INFORMATION
SUPPORT SYSTEM (IISS)
Volume V - Common Data Model Subsystem
Part 3 - CDM1, IDEF1 Model of the CDM

General Electric Company
Production Resources Consulting
One River Road
Schenectady, New York 12345

Final Report for Period 22 September 1980 - 31 July 1985

November 1985

Approved for public release; distribution is unlimited.

PREPARED FOR:

MATERIALS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AFB, OH 45433-6533

87 12 117

NOTICE

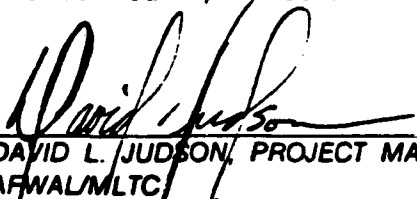When Government drawings. specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated. furnished, or in any way supplied the said drawings. specifications. or other data. is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS. it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

DAVID L. JUDSON, PROJECT MANAGER
AFWAL/MLTC
WRIGHT PATTERSON AFB OH 45433

DATE 5 Aug 1986

FOR THE COMMANDER:

GERALD C. SHUMAKER, BRANCH CHIEF
AFWAL/MLTC
WRIGHT PATTERSON AFB OH 45433

DATE 7 Aug 86

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE		Approved for public release; distribution is unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S) AFWAL-TR-86-4006 Vol V, Part 5

6a. NAME OF PERFORMING ORGANIZATION General Electric Company Production Resources Consulting	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION AFWAL/MLTC
6c. ADDRESS (City, State and ZIP Code) 1 River Road Schenectady, NY 12345		7b. ADDRESS (City, State and ZIP Code) WPAFB, OH 45433-6533
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Materials Laboratory Air Force Systems Command, USAF	8b. OFFICE SYMBOL (If applicable) AFWAL/MLTC	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33615-80-C-5155

8c. ADDRESS (City, State and ZIP Code) Wright-Patterson AFB, Ohio 45433	10. SOURCE OF FUNDING NOS.			
	PROGRAM ELEMENT NO. 78011F	PROJECT NO. 7500	TASK NO. 62	WORK UNIT NO. 01

11. TITLE (Include Security Classification)
(See Reverse)

12. PERSONAL AUTHOR(S)
Rollins, Dale

13a. TYPE OF REPORT Final Technical Report	13b. TIME COVERED 22 Sept 1980 - 31 July 1985	14. DATE OF REPORT (Yr., Mo., Day) 1985 November	15. PAGE COUNT 106

16. SUPPLEMENTARY NOTATION

ICAM Project Priority 6201

The computer software contained herein are theoretical and/or references that in no way reflect Air Force-owned or -developed computer software.

17.	COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB GR.	
1308	0905		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Common Data Model Processor (CDMP) is a mechanism by which application programs can retrieve and update data without knowing where or how the data are stored. The CDMP relies on three types of schemas (conceptual, external, and internal) and on two types of schema mappings (conceptual-to-external and conceptual-to-internal). A schema is a description of data from a particular viewpoint. A mapping is a description of the correspondence between elements of one schema and those of another. The schemas and mappings are stored in a database, called the Common Data Model (CDM), where they are available to the CDMP software. CDM1 is an IDEF1 model of the data requirements of the CDMP. It describes the entity, attribute, and relation classes that compose these schemas and mappings. It is also the basis for the design of the CDM database.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson	22b. TELEPHONE NUMBER (Include Area Code) 513-255-6976	22c. OFFICE SYMBOL AFWAL/MLTC

DD FORM 1473, 83 APR   EDITION OF 1 JAN 73 IS OBSOLETE.

11. Title

Integrated Information Support System (IISS)
Vol V - Common Data Model Subsystem
Part 3 - CDM1, IDEF1 Model of the CDM

The Integrated Information Support System is a test computing environment used to investigate and demonstrate and test the concepts of information management and information integration in the contexts of Aerospace Manufacturing. Specifically, IISS addresses the problems of integration of data resident on heterogeneous databases supported by heterogeneous computers, interconnected via a Local Area Network. A common Data Model is maintained and provides the mechanism required to integrate the data.

→ to 1473

PREFACE

This model of the CDM covers the work performed under Air Force Contract F33615-80-C-5155 (ICAM Project 6201). This contract is sponsored by the Materials Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Gerald C. Shumaker, ICAM Program Manager, Manufacturing Technology Division, through Project Manager, Mr. David Judson. The Prime Contractor was Production Resources Consulting of the General Electric Company, Schenectady, New York, under the direction of Mr. Alan Rubenstein. The General Electric Project Manager was Mr. Myron Hurlbut of Industrial Automation Systems Department, Albany, New York.

Certain work aimed at improving Test Bed Technology has been performed by other contracts with Project 6201 performing integrating functions. This work consisted of enhancements to Test Bed software and establishment and operation of Test Bed hardware and communications for developers and other users. Documentation relating to the Test Bed from all of these contractors and projects have been integrated under Project 6201 for publication and treatment as an integrated set of documents. The particular contributors to each document are noted on the Report Documentation Page (DD1473). A listing and description of the entire project documentation system and how they are related is contained in document FTR620100001, Project Overview.

The subcontractors and their contributing activities were as follows:

TASK 4.2

Subcontractors	Role
Boeing Military Aircraft Company (BMAC)	Reviewer
D. Appleton Company (DACOM)	Responsible for IDEF support, state-of-the-art literature search
General Dynamics/ Ft. Worth	Responsible for factory view function and information models

iii

Subcontractors	Role
Illinois Institute of Technology	Responsible for factory view function research (IITRI) and information models of small and medium-size business
North American Rockwell	Reviewer
Northrop Corporation	Responsible for factory view function and information models
Pritsker and Associates	Responsible for IDEF2 support
SofTech	Responsible for IDEF0 support

TASKS 4.3 - 4.9 (TEST BED)

Subcontractors	Role
Boeing Military Aircraft Company (BMAC)	Responsible for consultation on applications of the technology and on IBM computer technology.
Computer Technology Associates (CTA)	Assisted in the areas of communications systems, system design and integration methodology, and design of the Network Transaction Manager.
Control Data Corporation (CDC)	Responsible for the Common Data Model (CDM) implementation and part of the CDM design (shared with DACOM).
D. Appleton Company (DACOM)	Responsible for the overall CDM Subsystem design integration and test plan, as well as part of the design of the CDM (shared with CDC). DACOM also developed the Integration Methodology and did the schema mappings for the Application Subsystems.

Subcontractors	Role
Digital Equipment Corporation (DEC)	Consulting and support of the performance testing and on DEC software and computer systems operation.
McDonnell Douglas Automation Company (McAuto)	Responsible for the support and enhancements to the Network Transaction Manager Subsystem during 1984/1985 period.
On-Line Software International (OSI)	Responsible for programming the Communications Subsystem on the IBM and for consulting on the IBM.
Rath and Strong Systems Products (RSSP) (In 1985 became McCormack & Dodge)	Responsible for assistance in the implementation and use of the MRP II package (PIOS) that they supplied.
SofTech, Inc.	Responsible for the design and implementation of the Network Transaction Manager (NTM) in 1981/1984 period.
Software Performance Engineering (SPE)	Responsible for directing the work on performance evaluation and analysis.
Structural Dynamics Research Corporation (SDRC)	Responsible for the User Interface and Virtual Terminal Interface Subsystems.

Other prime contractors under other projects who have contributed to Test Bed Technology, their contributing activities and responsible projects are as follows:

Contractors	ICAM Project	Contributing Activities
Boeing Military Aircraft Company (BMAC)	1701, 2201, 2202	Enhancements for IBM node use. Technology Transfer to Integrated Sheet Metal Center (ISMC)

Contractors	ICAM Project	Contributing Activities
Control Data Corporation (CDC)	1502, 1701	IISS enhancements to Common Data Model Processor (CDMP)
D. Appleton Company (DACOM)	1502	IISS enhancements to Integration Methodology
General Electric	1502	Operation of the Test Bed and communications equipment.
Hughes Aircraft Company (HAC)	1701	Test Bed enhancements
Structural Dynamics Research Corporation (SDRC)	1502, 1701, 1705	IISS enhancements to User Interface/Virtual Terminal Interface (UI/VTI)
Systran	1502	Test Bed enhancements. Operation of Test Bed.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

TABLE OF CONTENTS (Continued)

SECTION 1

SCOPE

1.1 Identification

This document contains an IDEF1 data model of the information associated with the "TO-BE" Common Data Model (CDM) Subsystem of the Integrated Information Support System (IISS). Section 1 gives a brief overview of the CDM Subsystem. Section 2 identifies the other associated ICAM documents. The CDM1 model is depicted in a series of Function View diagrams shown in Section 3. Definitions and indexes for entities and attributes are contained in Sections 4 and 5 respectively.

1.2 Overview

The purpose of the model is to serve as a guide for the development of the CDM Database and the CDM Processor (CDMP). These components of the CDM Subsystem are described in the following paragraphs. However, not all aspects of the CDM1 model are implemented by the current IISS development specifications and demonstration software.

The CDM Database

The CDM database is the data dictionary of the IISS. It captures knowledge of the locations, characteristics, and inter-relationships of all shared data in the system. The most significant feature of the CDM database is that it implements the ANSI/X3/SPARC concepts of the three-schema approach to data management. The three types of schemas are the conceptual schema (CS), the internal schemas (IS), and the external schemas (ES).

The conceptual schema describes a neutral, integrated view of the shared data resource. There is one conceptual schema in an enterprise. It is independent of physical database structures and boundaries and is neutral to biases of individual applica- tions. Each external schema represents a user or application view of data. Each internal schema represents the logical structure of a local DBMS whether hierarchical, network, or relational.

The CDM database, itself is implemented as a relational database, which presently resides on a VAX 11/780 computer. It

is accessed by the CDMP at compile-time to generate appropriate
local DBMS calls against internal schemas to process a user's
NDML request against an external schema.

The CDM Processor

The CDMP is the distributed database manager of the IISS.
It builds on top of local DBMS services to provide data access.
The CDMP plays both a compile-time and a run-time role in the
processing of transactions. The compile-time component is
called the CDMP Precompiler. The run-time components are called
the CDMP Distributed Request Supervisor (DRS) and the CDMP
Aggregator.

The CDMP Precompiler performs the following functions for
each data request:

1. Parses the request,
2. Transforms the request from an external schema access
   to a conceptual schema access,
3. Decomposes the request into subrequests, each of which
   accesses one internal schema,
4. Determines an appropriate access path for each
   subrequest and generates code that can be processed by
   the pertinent local DBMS,
5. Generates code to transform any data to be extracted
   from local databases from internal to conceptual
   schema format (this code is called a Request
   Processor Packet or RPP),
6. Generates code to transform any data results from
   conceptual to external schema format (this code is
   called a C/E Transformer or CEX), and
7. Generates code to invoke appropriate RPPs and CEXs at
   run-time, via calls to the NTM Subsystem.

The CDMP Precompiler accesses the CDM database to find
metadata for the interschema transforms and integrity
constraints for update requests.

After successful precompilation of a user's program, which
contains imbedded data requests in an SQL-like language called
the Neutral Definition Manipulation Language (NDML), the CDMP
has produced the following code modules:

1. Modified user program, which now contains calls to
   the NTM, which will activate appropriate processes at
   run-time.

2. One Request Processor (RP) per DBMS that manages data to be accessed by the user program. Each RP contains one or more RPPs.
3. One Conceptual-to-External Transformer (CEX), which will deliver query results to the modified user program at run-time.

The CDMP Distributed Request Supervisor (DRS) has responsibility for scheduling and coordinating the various subrequests of user transactions. The DRS uses request graphs produced by the CDMP Precompiler to determine which operations are to be performed where. The DRS also uses knowledge of communications costs and intermediate result volumes in its algorithm for scheduling RPPs. Request Processors always deliver results as relations. The relations are operated upon by the Aggregators.

Aggregators are called to perform single functions, e.g., a union or a join, on two sets of data, each of which exists in a single sequential file. These data sets are the results of an RPP or processing by another Aggregator. An Aggregator always deals with data in conceptual schema format.

CDM1 Overview

CDM1 is a semantic data model of the IISS metadata, i.e. a semantic model of data about data. It was built using the IDEF1 data modeling approach with some minor extensions. (These extensions along with others, have now been refined and formalized in the IDEF1 - Extended Manual). Part of the metadata modeled by CDM includes the IISS Conceptual Schema View of manufacturing data, which is itself represented by an IDEF1 model.

The conceptual schema portion of the CDM1 model is related to portions that describe internal and external schemas. An internal schema describes a local database structure in just enough detail to give the CDMP adequate information to generate code that can be processed by the pertinent local DBMS. The mappings between the conceptual schema metadata and the internal schema metadata are not simple, because one of the requirements of the IISS is that it provide integration of data in existing databases. IISS does not have the luxury of supporting only certain clean database structures. It is very likely that an attribute may be represented by one or more data files, which may be in different databases and even on different computers, or represented by relationships between record types.

An external schema describes the portion of the conceptual schema that is within the purview of a user or application. An external schema that is equivalent to a view in the relational model. The conceptual-to-external schema mapping part of the CDM1 is straightforward. The present implementation of the CDM subsystem supports any external schema that can be formed by joining conceptual schema entities and selecting attributes.

Thus, the CDM1 model is a semantic data model that describes the logical structure of the CDM database. The CDM1 represents the conceptual schema, the internal schemas and their mappings from the conceptual schema, and the external schemas and their mappings from the conceptual schema.

SECTION 2

DOCUMENTS

2.1 Applicable Documents

Related ICAM Documents included:

UM620141001	CDM Administrator's Manual
PRM620141200	NDML Programmer's Reference Manual
UM620141100	Neutral Data Definition Language (NDDL) User's Guide
UM620141002	Information Modeling Manual - IDEF1 Extended
DS620141320	Data Aggreegator DS
DS620141310	Distributed Request Supervisor DS
DS620141200	NDML Precompiler DS

2.2 Terms and Abbreviations

APL	Attribute Pair List
AUC	Attribute Use Class
CDMP	Common Data Model Processor
CI	Configuration Item
CS	Conceptual Schema
DML	Data Manipulation Language
DRS	Distributed Request Supervisor (previously SS: Stager/Scheduler)
ES	External Schema
ICAM	Integrated Computer Aided Manufacturing
IS	Internal Schema

NDML	Neutral Data Manipulation Language
RP	Request Processor
RFT	Result Field Table
SDS	System Design Specification

SECTION 3

FUNCTION VIEW DIAGRAMS

The CDM1 Model is depicted in seventeen Function View Diagrams (FEO's) which are shown in Figures 3-1 through 3-17. Notes are attached to each diagram describing changes since the last verison.
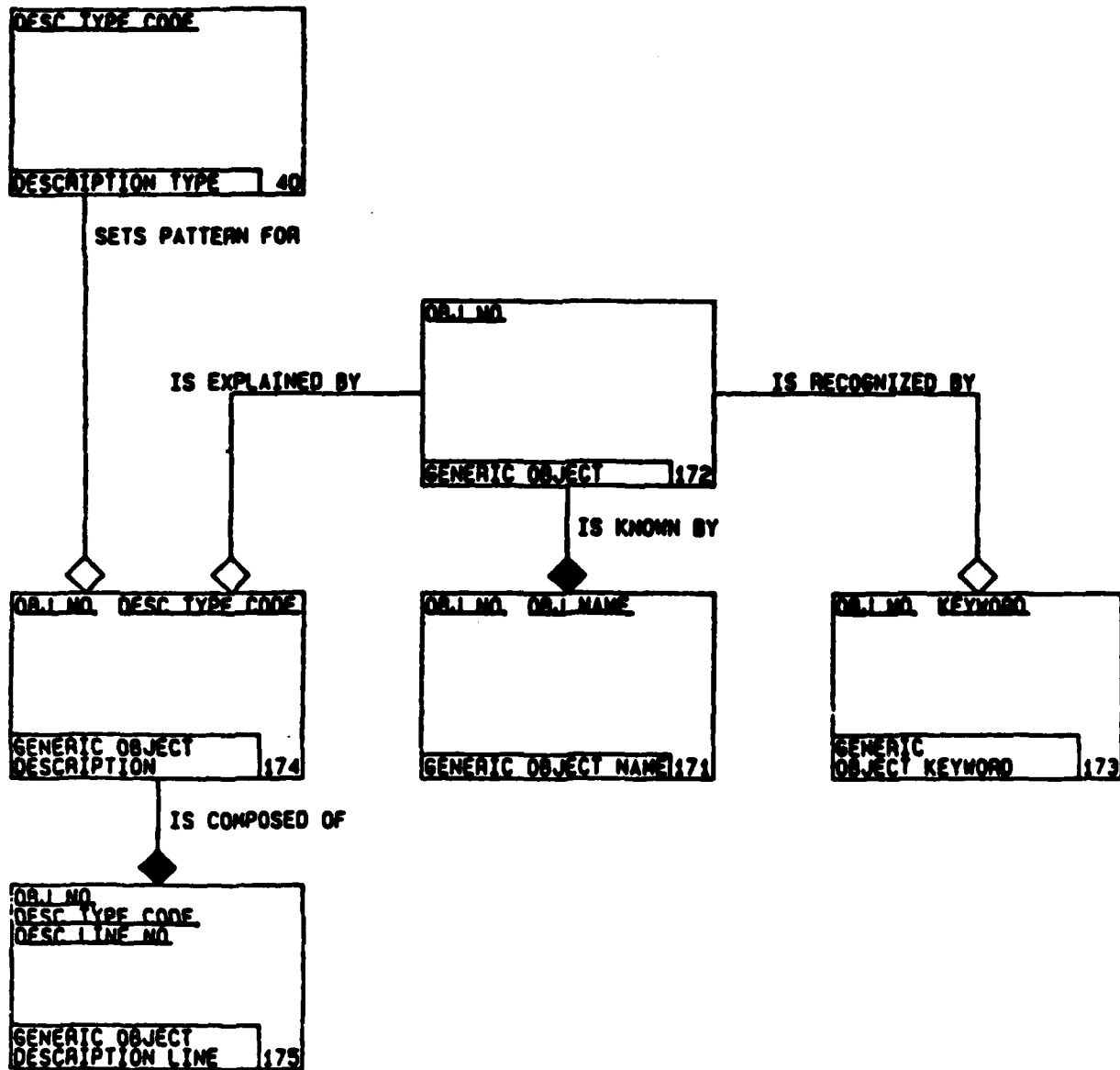
Figure 3-1.   F1: Generic Objects, Names, Aliases, Keywords,
               & Descriptions
               Changes to Diagram F1:
               No changes.
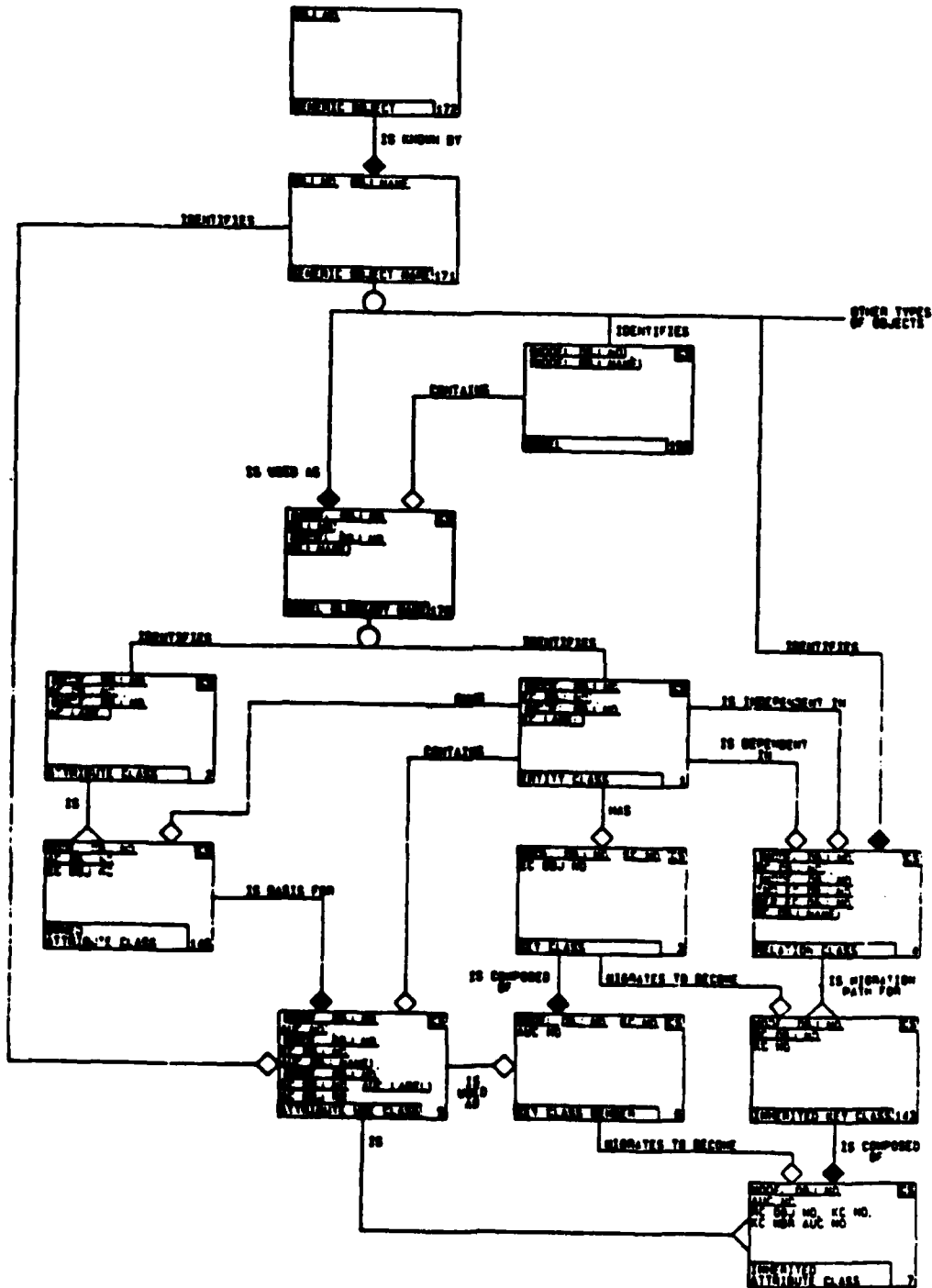
Figure 3-2. F2: Conceptual Schema (and IDEF1 Models) Changes
to Diagram F2:
No changes.
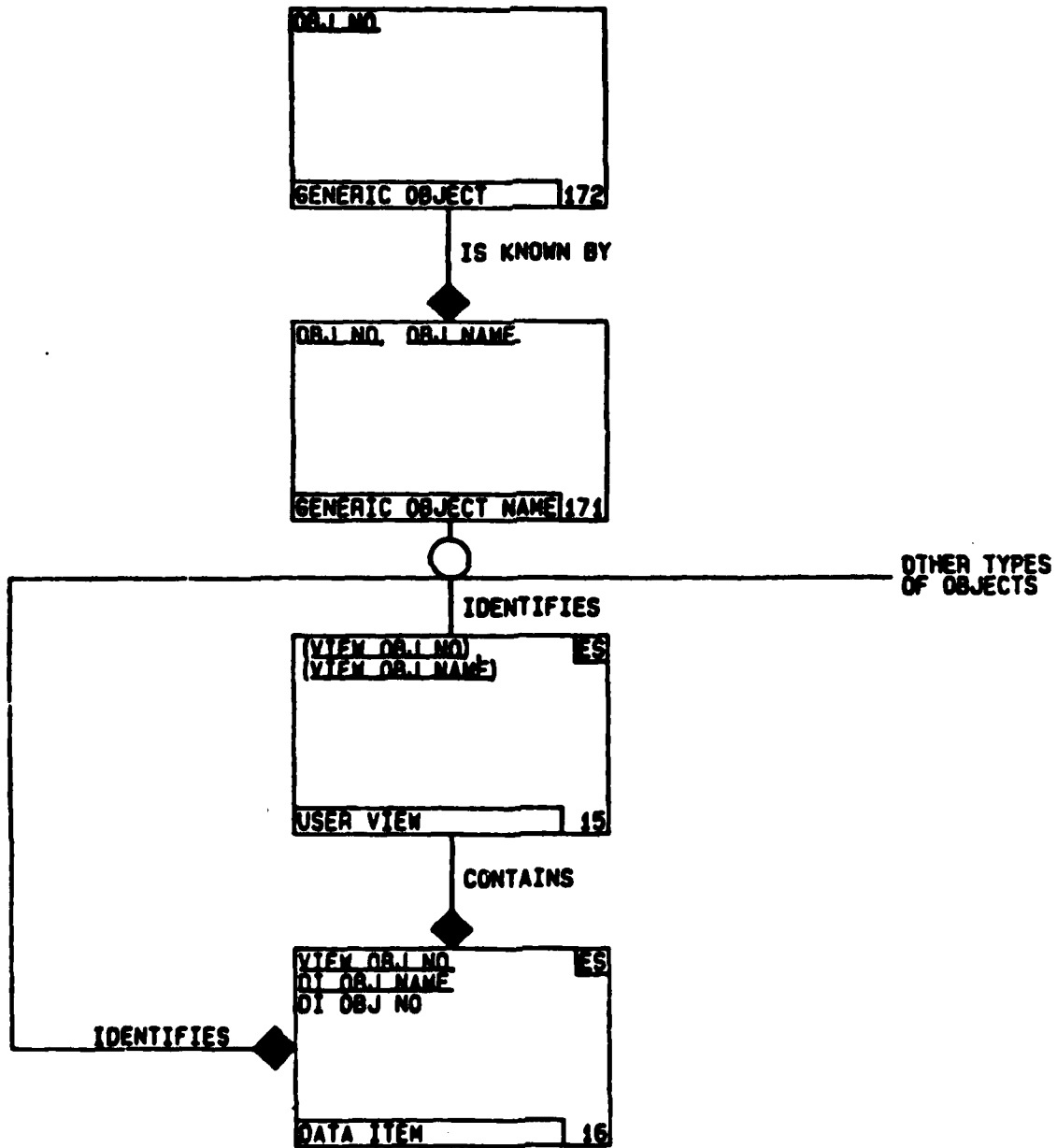
Figure 3-3. F3: External Schemas
             Changes to Diagram F3:
             No changes.
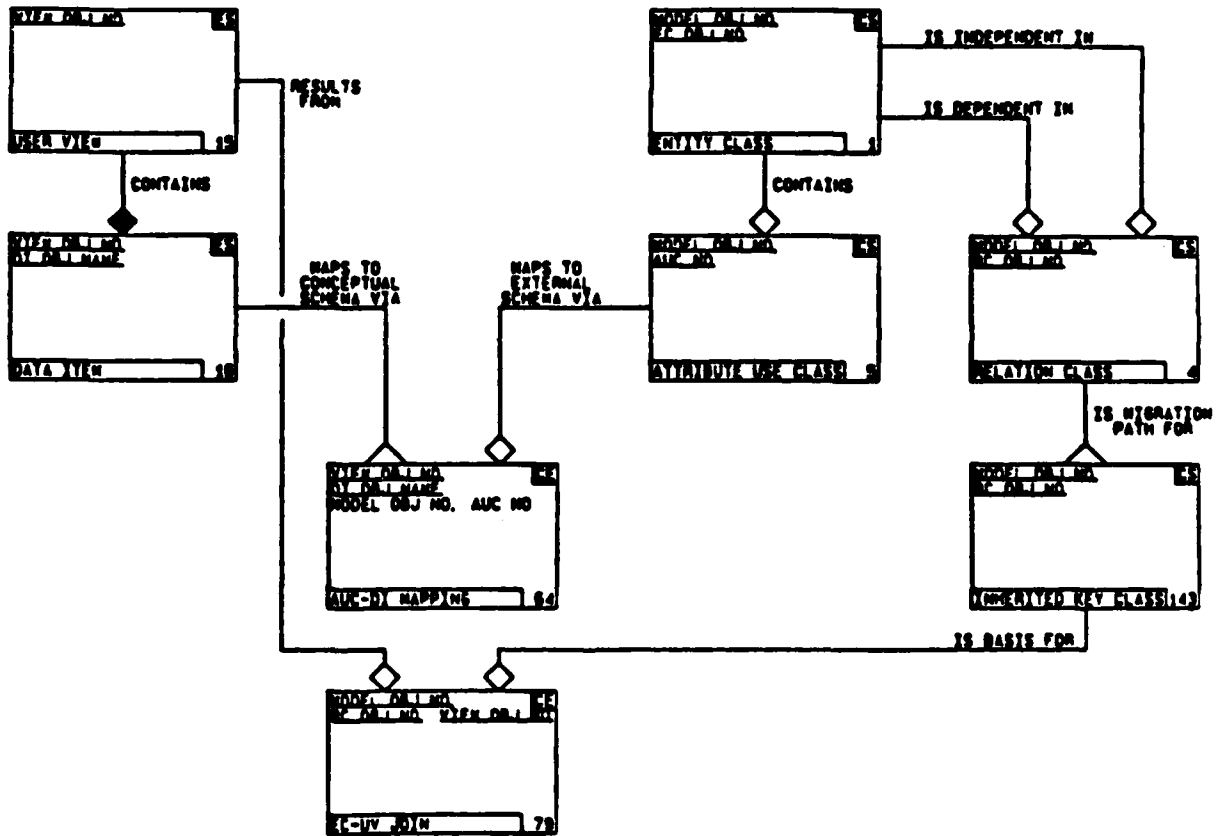
Figure 3-4.   F4:   CS-ES Mappings
             Changes to Diagram F4:
             No changes.
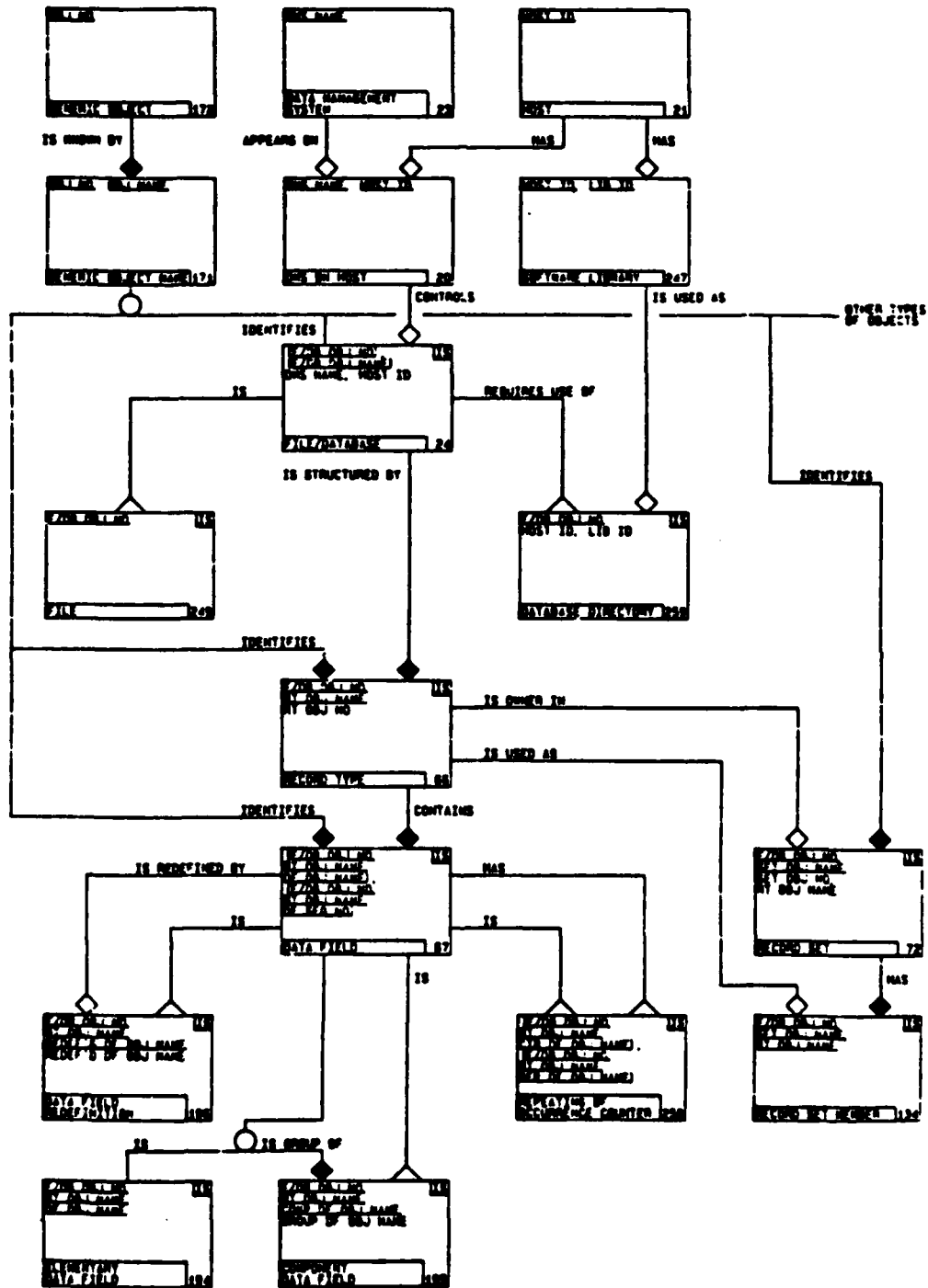
Figure 3-5.   F5:   Generic Internal Schemas

Changes to Diagram F5:

1. The Component Data Field Sequence Number in the
    Component Data Field entity class (E195) was replaced
    with a Data Field Sequence Number in the Data Field
    entity class (E67) so that all data fields will have
    sequence numbers, not just those that are components
    of group data fields. This will allow the
    Precompiler to put all data fields in the proper
    positions within the record type descriptions that it
    generates.

2. The Repeating Data Field Occurrence Counter entity
    class (E258) was added to represent a data field that
    indicates how many occurrences of another, repeating
    data field actually contain data values in each
    record instance.

3. The Database Directory entity class (E259) was added
    to represent directories that must be used to access
    databases.
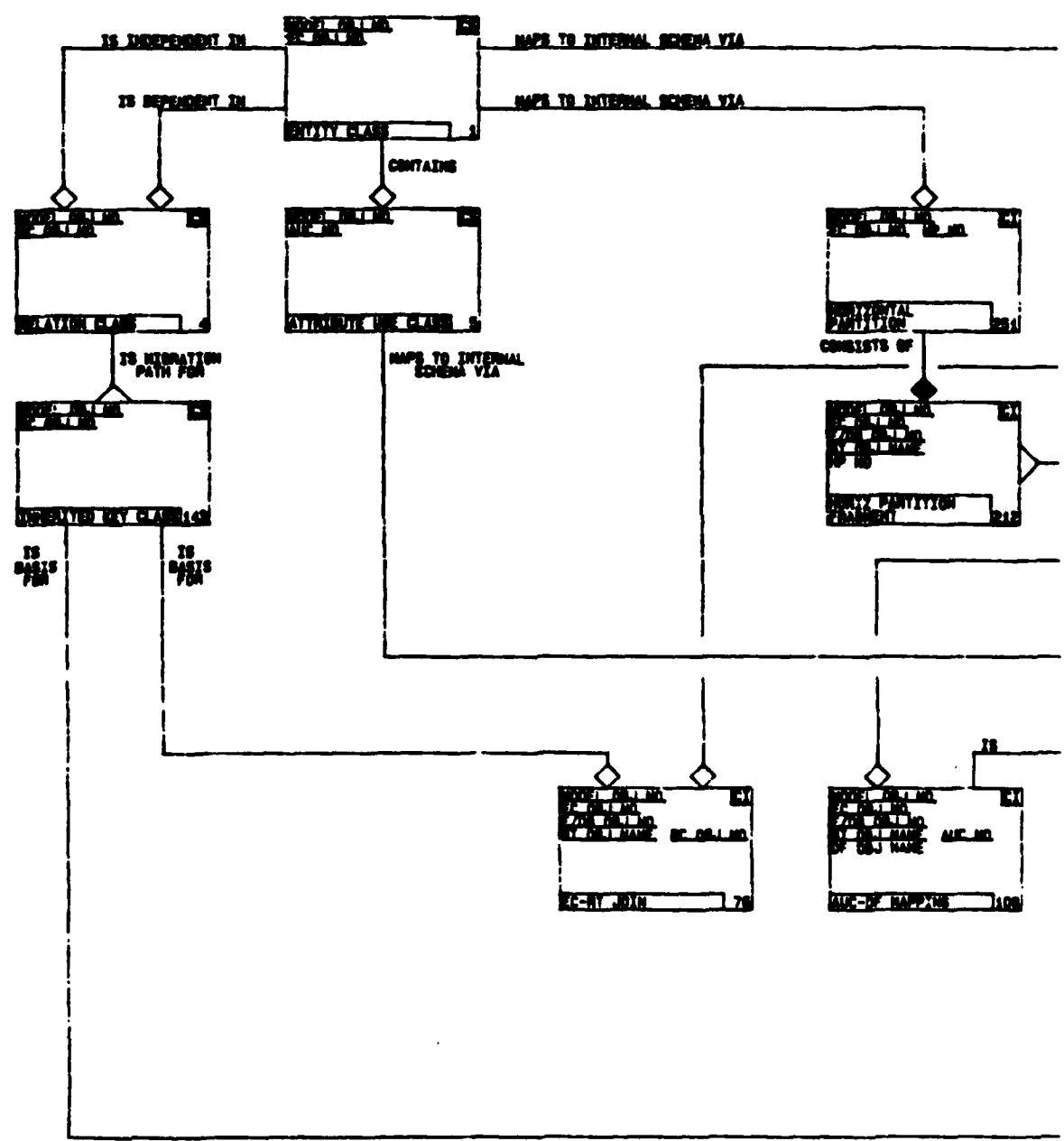
Figure 3-6.   F6:   CS-IS Mappings
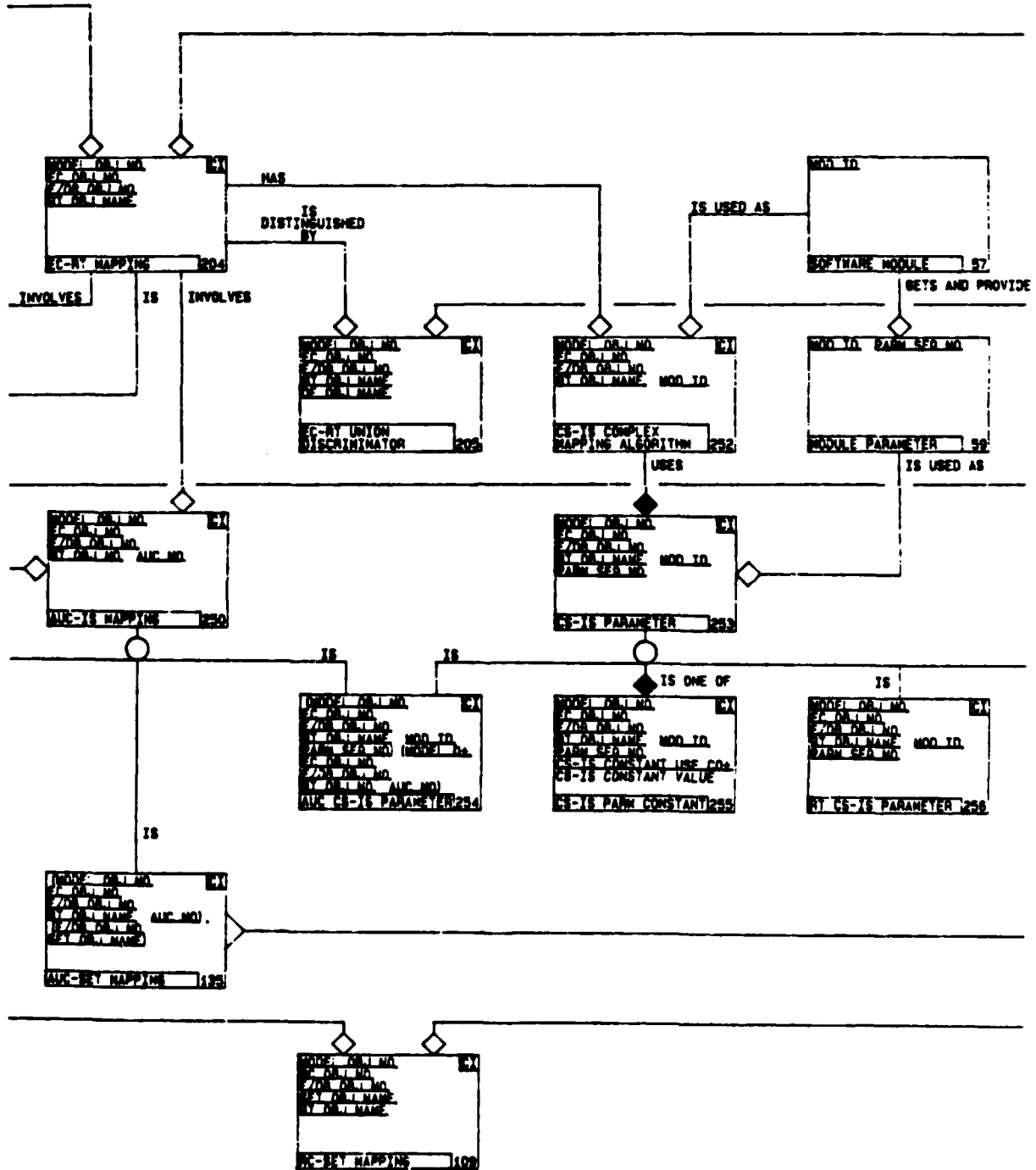
Figure 3-6.   F6:   CS-IS Mappings (Continued)
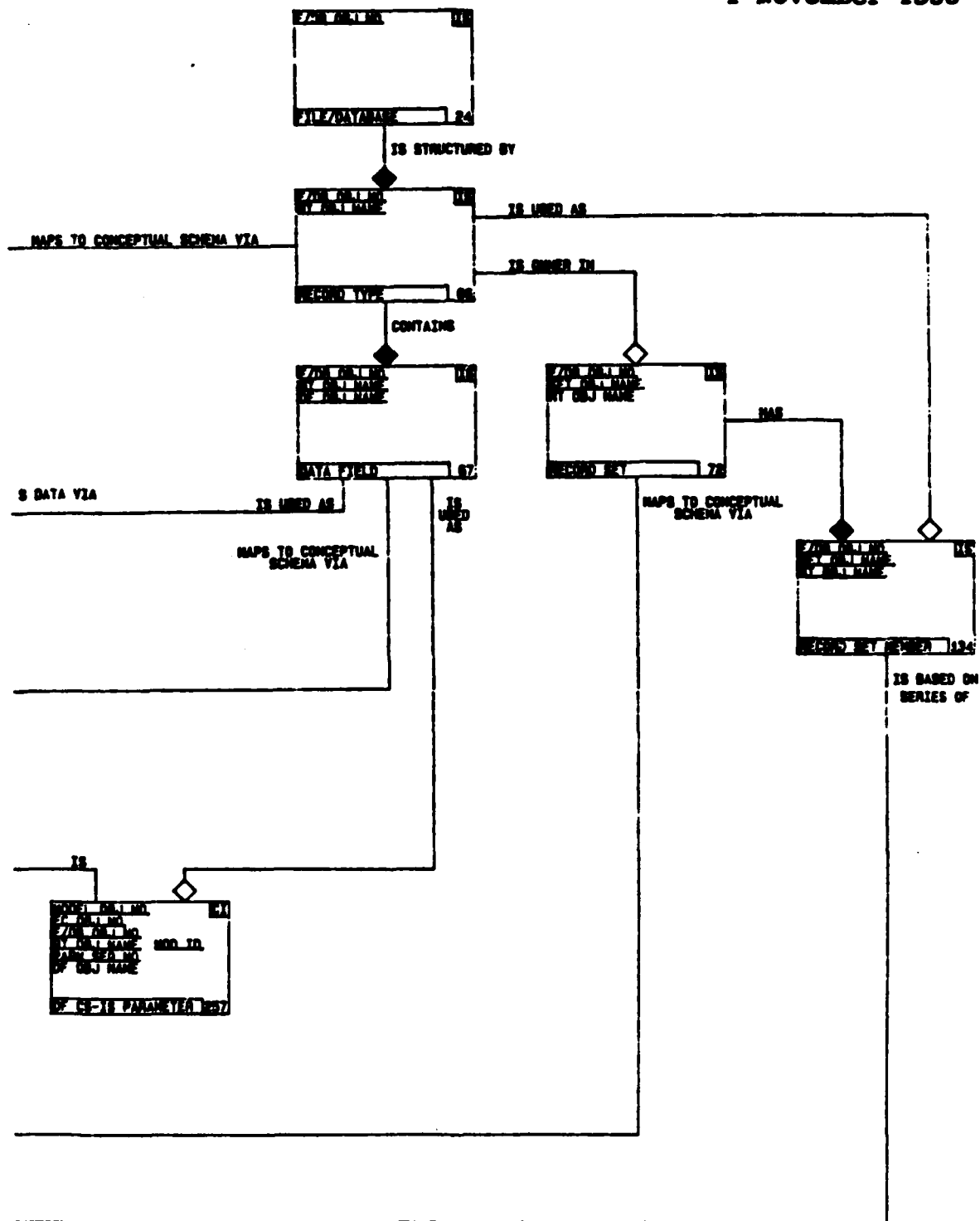
Figure 3-6. F6: CS-IS Mappings (Continued)

Changes to Diagram F6:

1. The Repeating Data Field Index entity class (E222) was replaced with a Repeating Data Field Index Indicator in the AUC-DF Mapping entity class (E108) because an attribute class is adequate for representing the mapping of an AUC to an index on a repeating data field.

2. The following entity classes were added to represent CS-IS complex mappings:

CS-IS Complex Mapping Algorithm	(E252)
CS-IS Parameter	(E253)
AUC CS-IS Parameter	(E254)
CS-IS Parameter Constant	(E255)
RT CS-IS Parameter	(E256)
DF CS-IS Parameter	(E257)

3. The AUC-IS Mapping entity class (E250) was added to represent the generalization of the following entity classes:

AUC-DF Mapping	(E108)
AUC-Set Mapping	(E135)
AUC CS-IS Parameter	(E254)

   The Preference Number was moved from the AUC-DF Mapping entity class to the AUC-IS Mapping entity class so that it could be applied to all types of AUC mappings.

4. The name of E212 was changed from Horizontal Partition to Horizontal Partition Fragment because it represents just one portion of the horizontal partition of an entity class, not all portions. A new entity class (E251) called Horizontal Partition was added to represent all the portions.

5. The relation class between Record Type (E66) and EC-RT Join (E76) was replaced with one between EC-RT Mapping (E204) and EC-RT Join so that, if a record type is results from both joins and a union, each join can be associated with the appropriate part of the union.

Figure 3-6. F6: CS-IS Mappings (Continued)
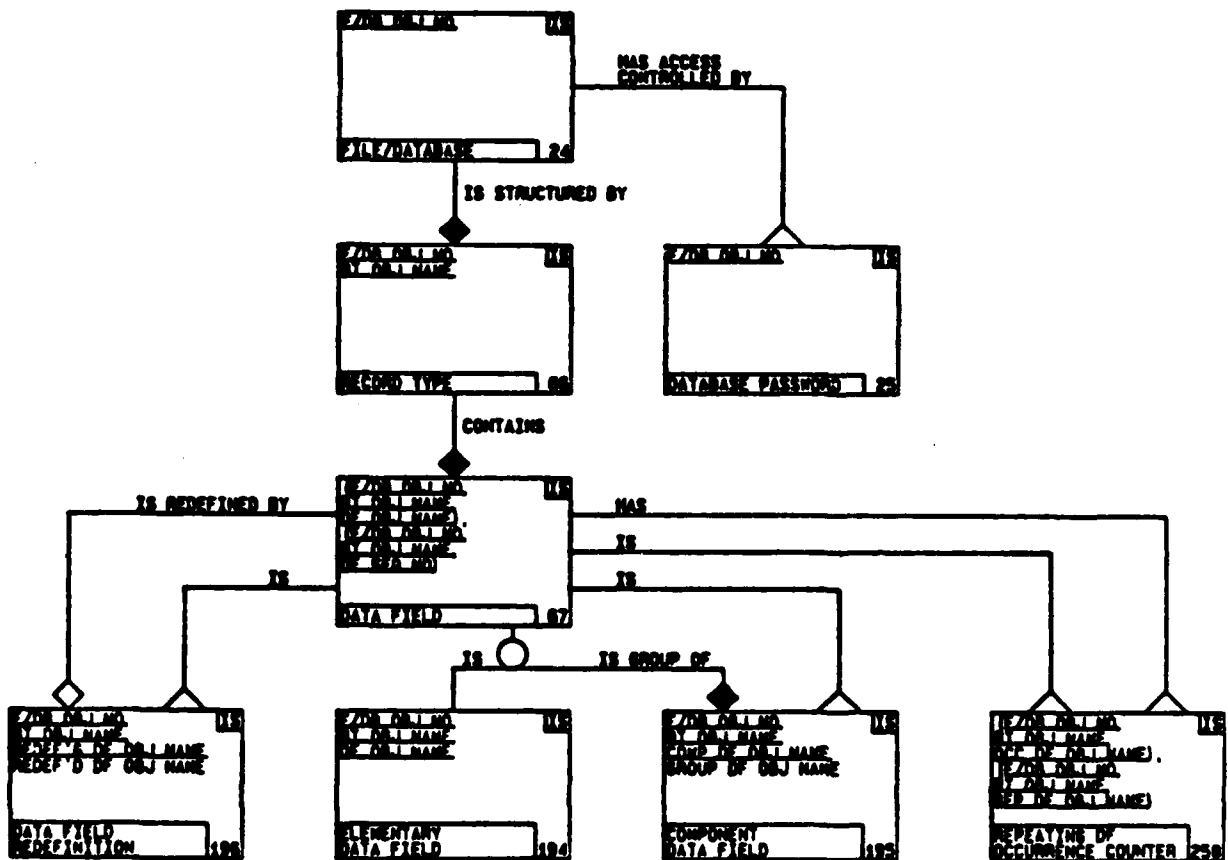
Figure 3-7. F7: Relational Internal Schemas (ORACLE and DB2)
Changes to Diagram F7:

1.   The following entity classes from diagram F5 were
     added to better depict what is available in this type
     of internal schema:

Data Field Redefinition	(E196)
Elementary Data Field	(E194)
Component Data Field	(E195)
Repeating DF Occurrence Counter	(E258)
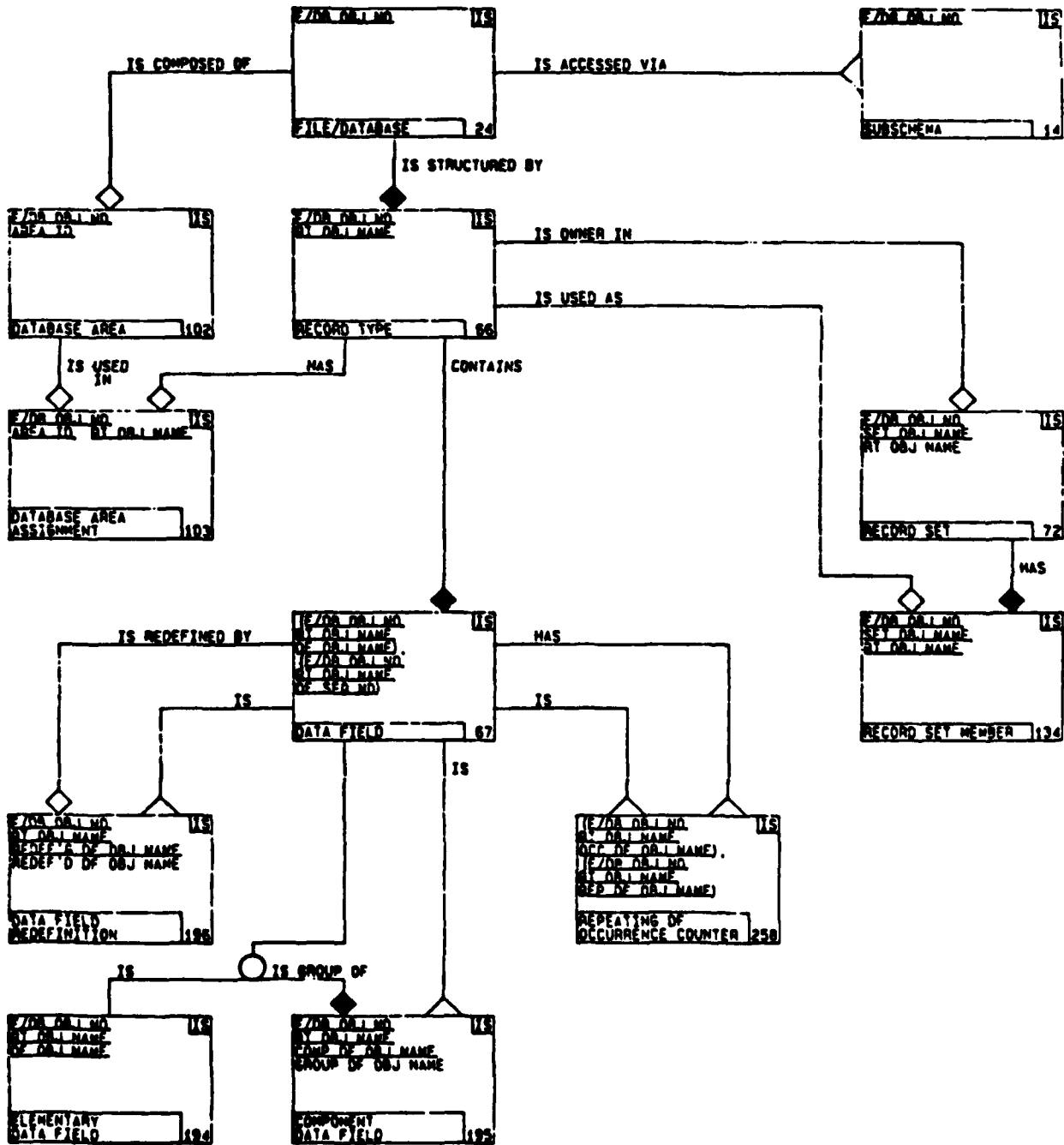
Figure 3-8. F8: CODASYL Internal Schemas

Changes to Diagram F6:

1. The following entity classes from diagram F5 were
    added to better depict what is available in this type
    of internal schema:

Data Field Redefinition	(E196)
Elementary Data Field	(E194)
Component Data Field	(E195)
Repeating DF Occurrence Counter	(E258)
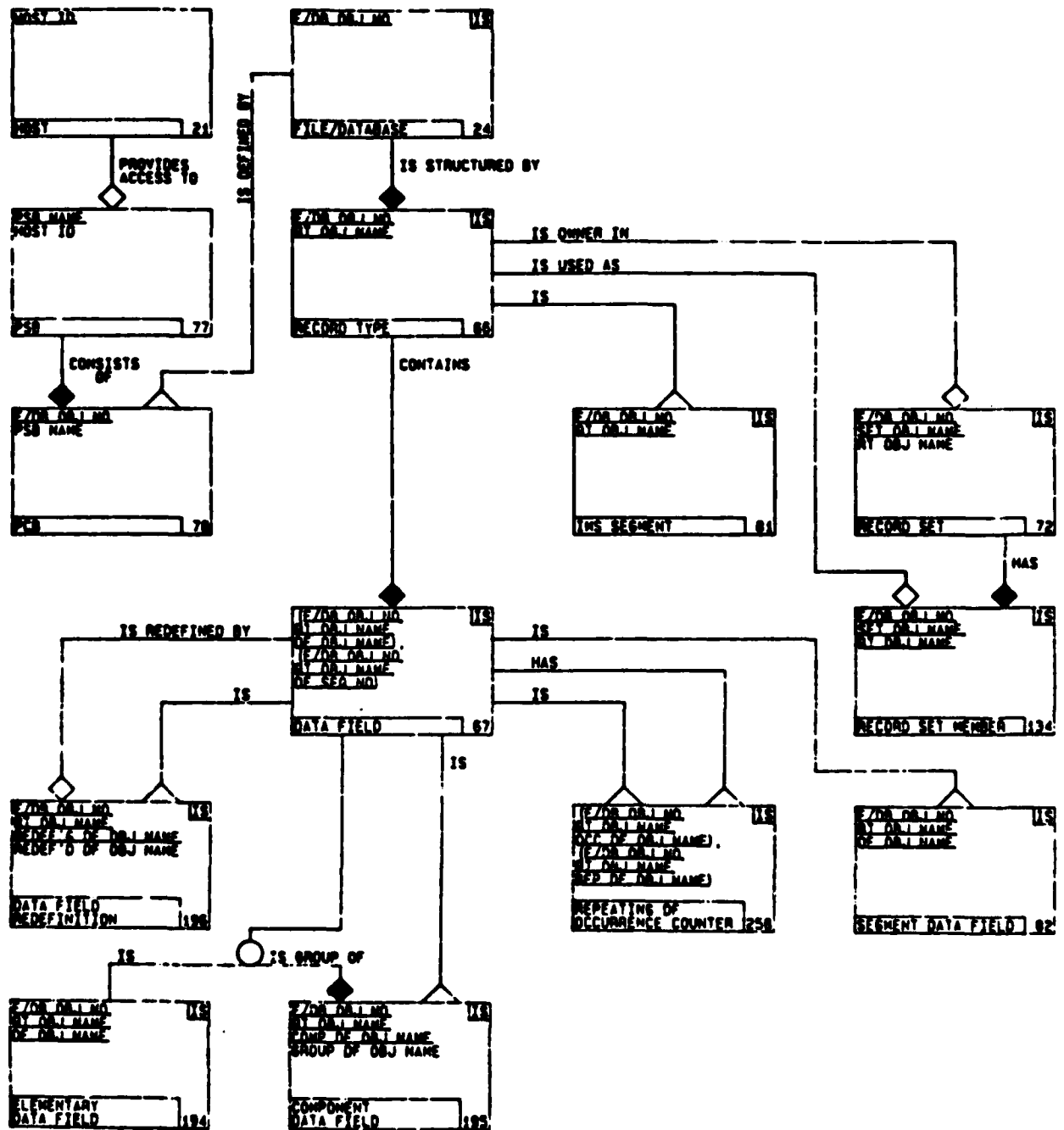
3-9. F9: IMS Internal Schemas

Changes to Diagram F9:

1. The following entity classes from diagram F5 were
    added to better depict what is available in this type
    of internal schema:

Data Field Redefinition	(E196)
Elementary Data Field	(E194)
Component Data Field	(E195)
Repeating DF Occurrence Counter	(E256)

2. The IMS Data Field Indicator in the Segment Data
    Field entity class (E82) was replaced with a DBMS-
    Accessible Data Field Indicator in the Data Field
    entity class (E67) because any type of database may
    contain data fields that are unknown to the DBMS.
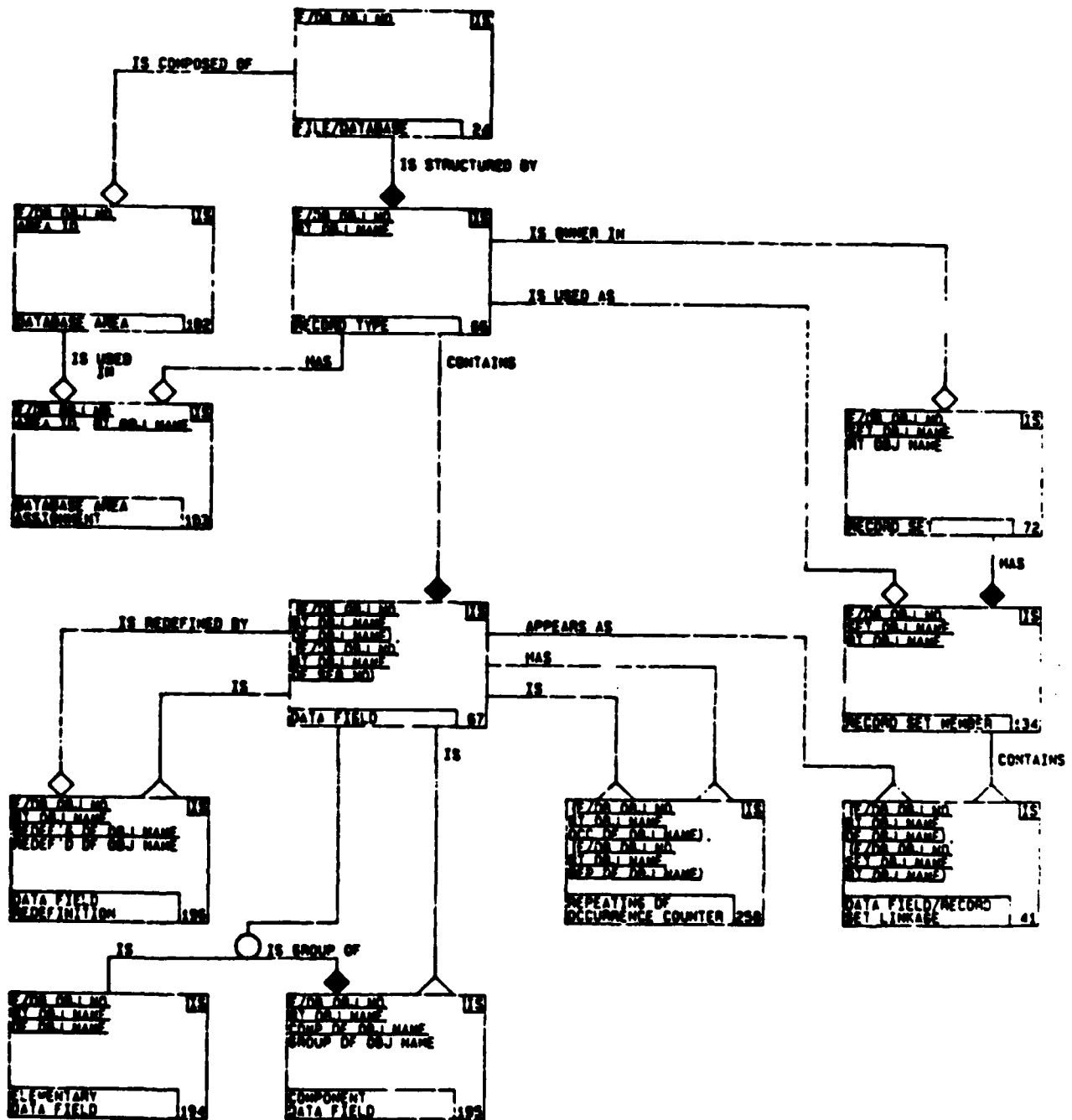
3-10. F10: TOTAL Internal Schemas

Changes to Diagram F10:

1. The following entity classes from diagram F5 were
    added to better depict what is available in this type
    of internal schema:

Data Field Redefinition	(E196)
Elementary Data Field	(E194)
Component Data Field	(E195)
Repeating DF Occurrence Counter	(E258)
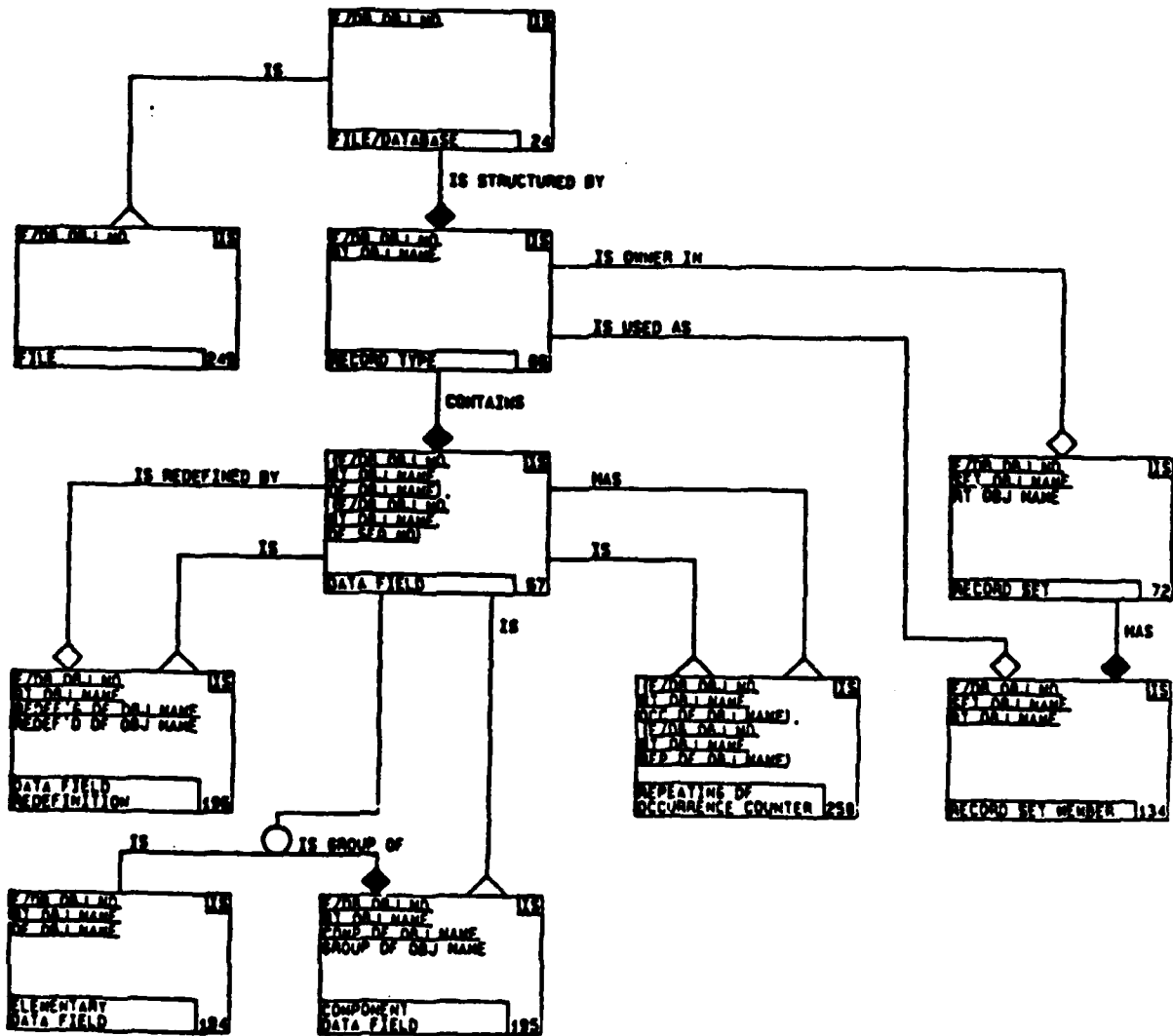
Figure 3-11. F11: VSAM and Flat File Internal Schemas

Changes to Diagram F11:

1. The following entity classes from diagram F5 were
    added to better depict what is available in this type
    of internal schema:

Data Field Redefinition	(E196)
Elementary Data Field	(E194)
Component Data Field	(E195)
Repeating DF Occurrence Counter	(E258)
Record Set	(E72)
Record Set Member	(E134)
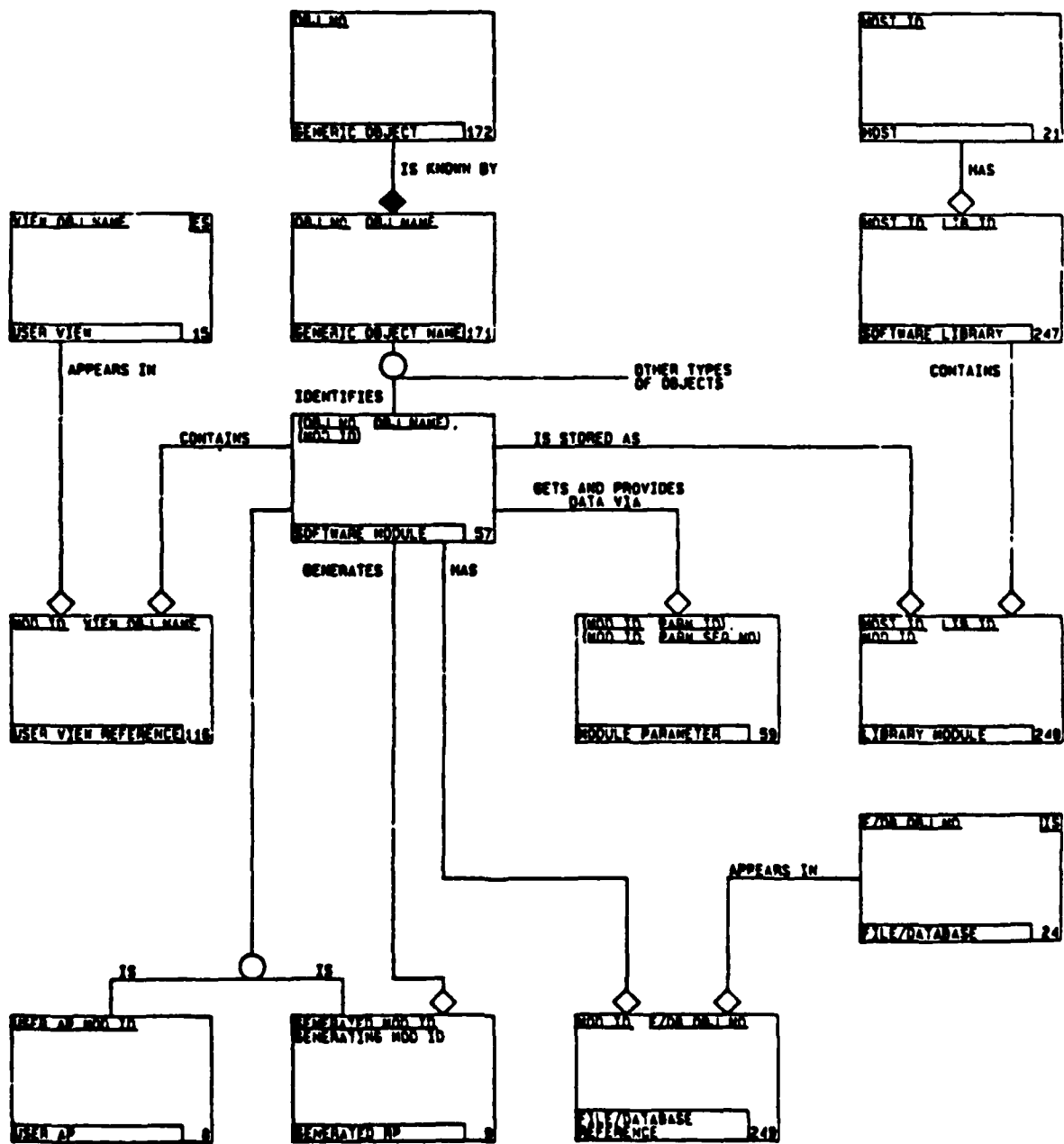
Figure 3-12. F12: Software Modules

Changes to Diagram F12:

1. A Parameter Sequence Number was added to the Module
   Parameter entity class (E59) so that parameters can
   be listed in the proper sequence in generated CALL
   statements.
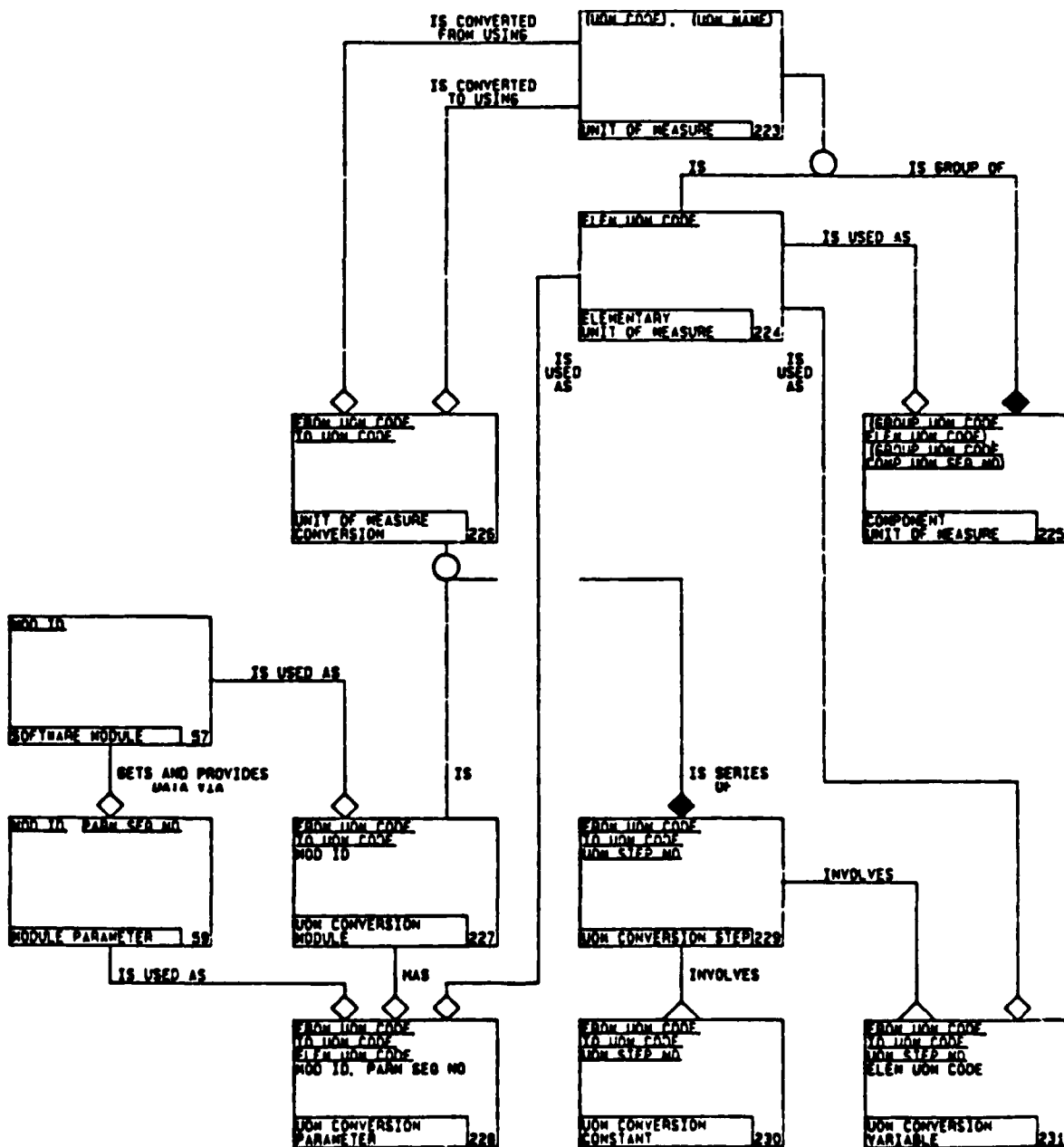
Figure 3-13.   F13:   Units of Measure
                Changes to Diagram F13:
                No changes.
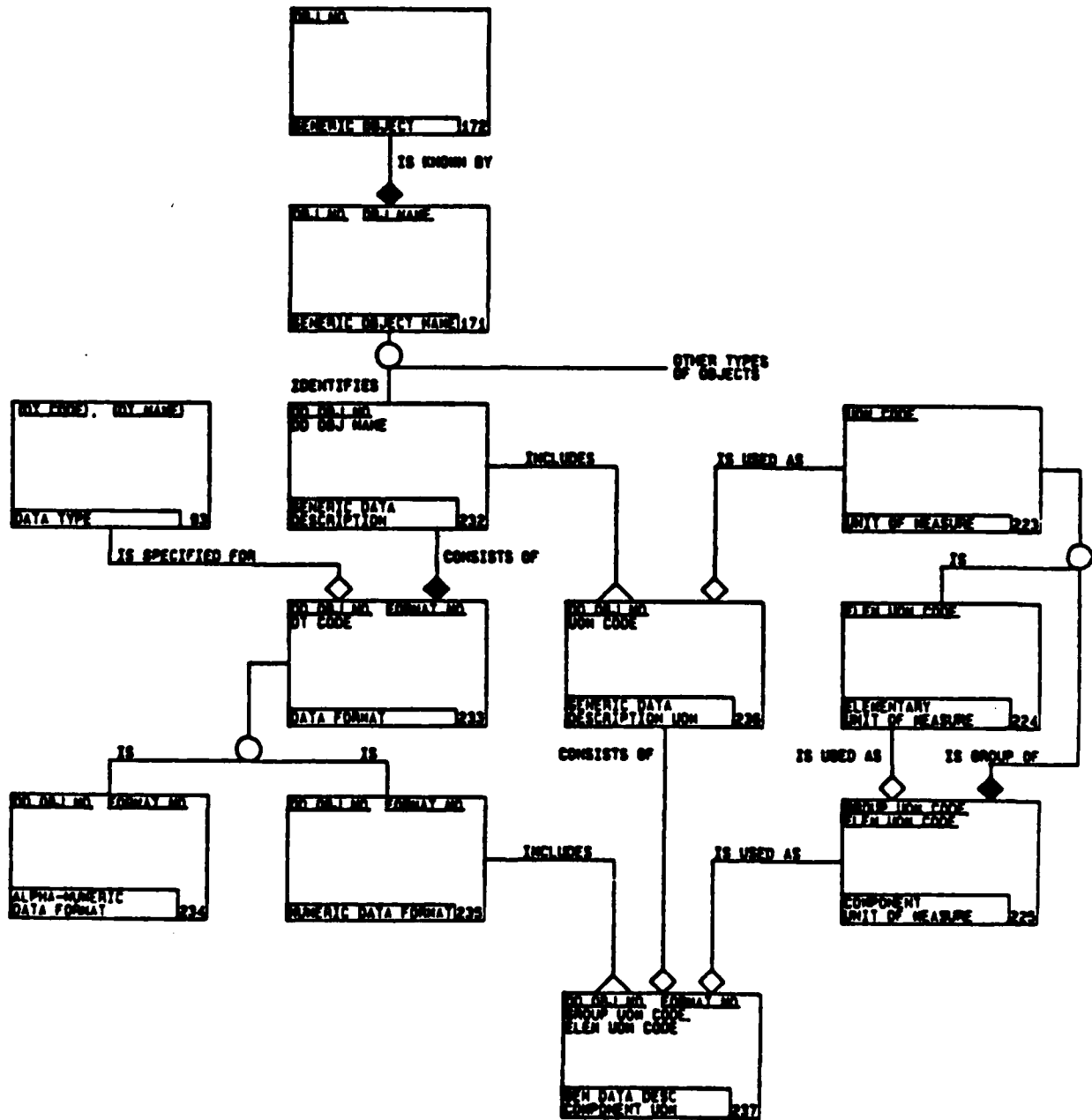
Figure 3-14.   F14:   Generic Data Descriptions

Changes to Diagram F14:

1. The diagram was corrected to show the Data Type Code in the Data Format entity class (E233) as nonkey rather than key.
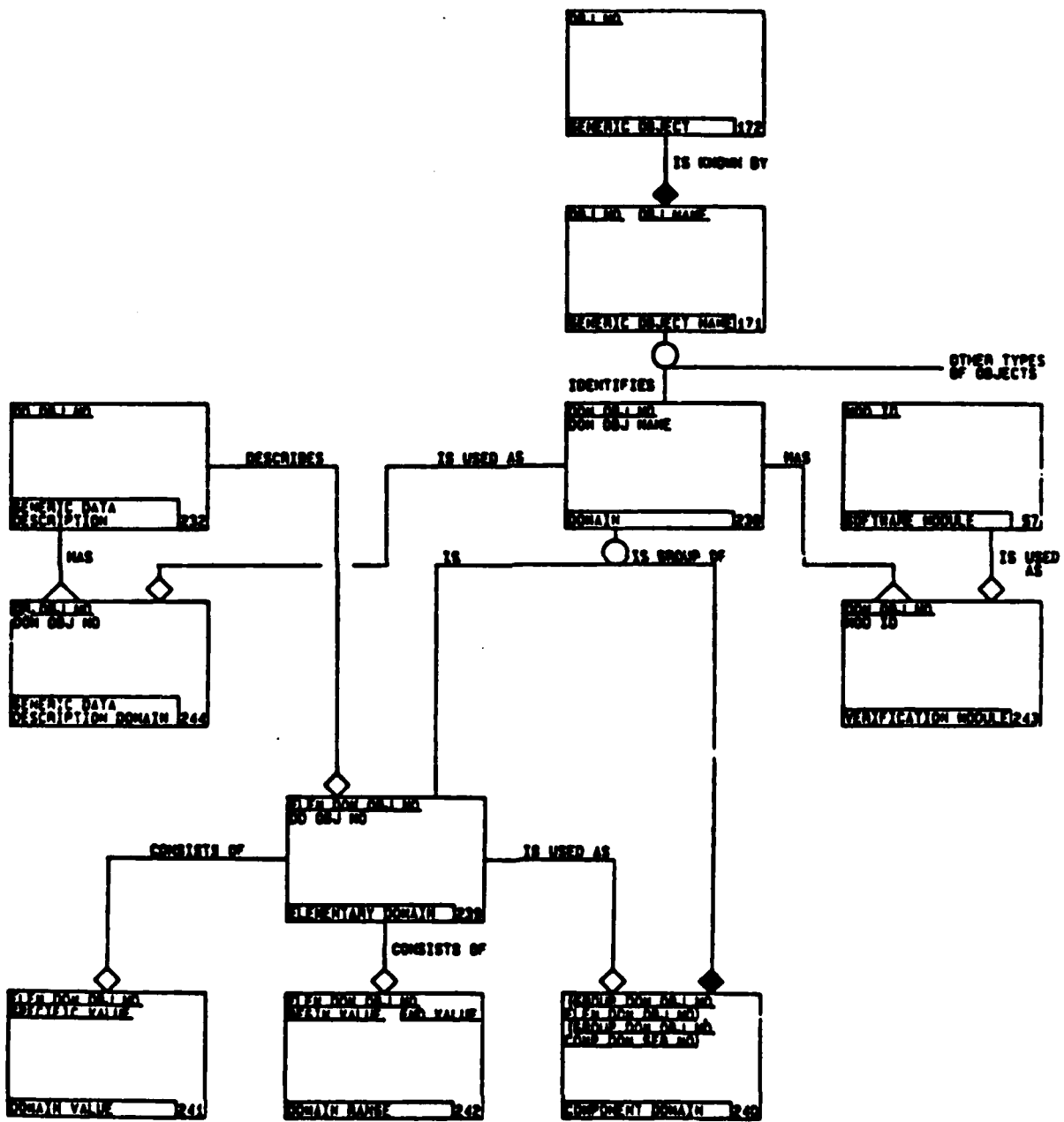
Figure 3-15.   F15:   Domains
Changes to Diagram F15:
No changes.
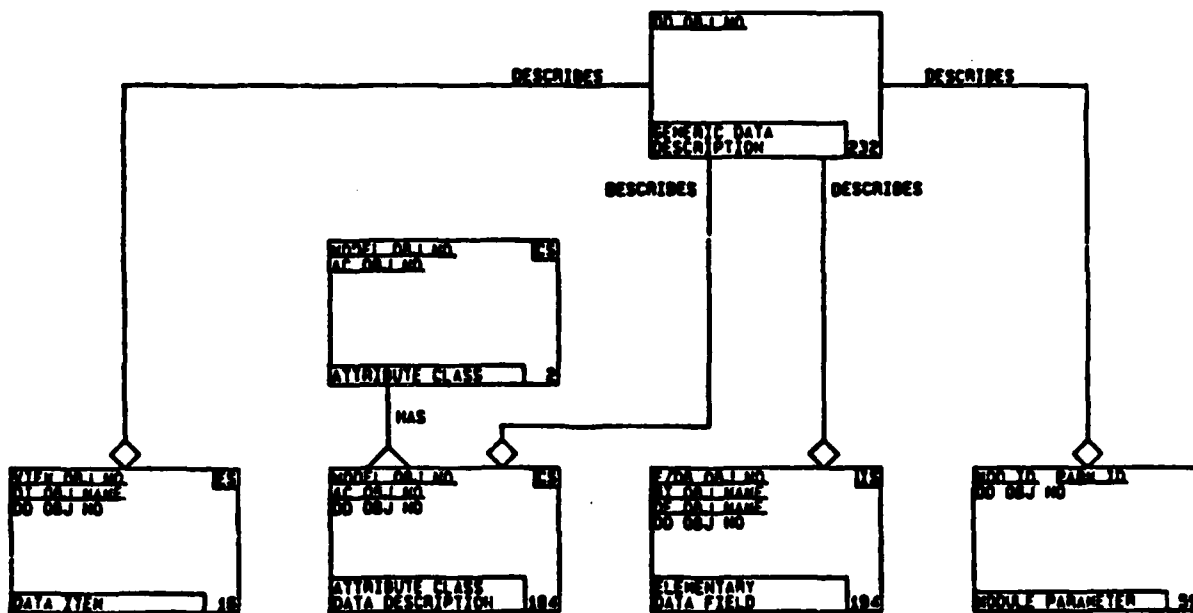
Figure 3-16.   F16:   Uses of Generic Data Descriptions
               Changes to Diagram F16:
               No changes.
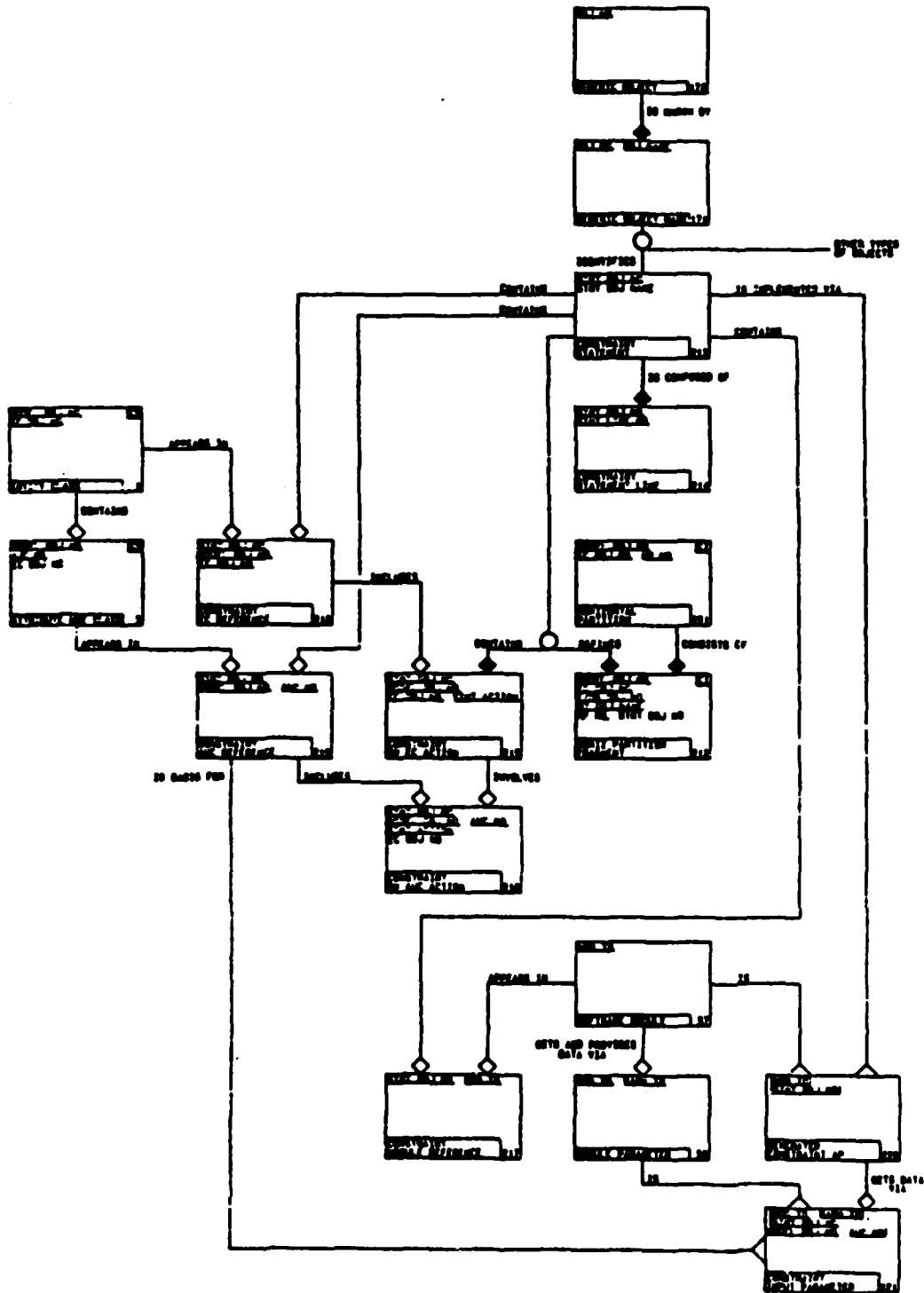
Figure 3-17. F17: Constraint Statements

Changes to Diagram F17:

1. The cardinality of the relation class between Generated Constraint AP (E220) and Constraint Input Parameter (E221) was changed from 1:1,M to 1:0,M so that generated constraint APs do not have to have any input parameters.

2. The name of E212 was changed from Horizontal Partition to Horizontal Partition Fragment to conform to a change in diagram F6. Also, the EC-RT Mapping entity class (E204) was replaced (in this diagram only) with the new Horizontal Partition entity class (E251).

SECTION 4

ENTITY CLASS DOCUMENTATION

4.1 CDM1 Entity Class Glossary

Note:
The entity classes in this glossary are in alphabetic
sequence by entity class name. Entity class numbers appear
in parentheses following the names. Entity class labels
that are different than the names appear with the numbers.
If a label is the same as a name, the label is omitted.

Alpha-Numeric Data Format (E234)

A data format for values that can contain characters other
than numerals (0-9). Numerals may be permitted also.

Attribute Class (E2)

A collection of all the same kind of attributes, i.e.,
those that have the same meaning. An attribute is a
characteristic or fact about an entity. An attribute consists
of a name (e.g., employee hire date) and a value (e.g., 15
August 1980). An attribute value may be:

A. Nondivisible (e.g., state name)

B. Divisible, i.e., a concatenation of two or more other
     attribute values (e.g., part number formed by
     concatenating drawing number and material code).

C. Computed from one or more other attribute values (e.g.,
     age computed as current date minus birth date).

Attribute Class Data Description (E184)

A generic data description that applies to a particular
attribute class.

Attribute Use Class (E5)

A model attribute class that appears in a model entity class. Each attribute use class represents either an owned attribute class or an inherited attribute class.

Attribute Use Class Conceptual Schema/Internal Schema Mapping Parameter (E254, AUC CS-IS Parameter)

An attribute use class that is used as a CS-IS mapping parameter.

Attribute Use Class/Data Field Mapping (E108, AUC-DF Mapping)

Indicates that an attribute use class corresponds to a data field; i.e., that they have the same meaning and that the data field can be used to store values for the attribute use class.

Attribute Use Class/Data Item Mapping (E64, AUC-DI Mapping)

Indicates that an attribute use class corresponds to a data item; i.e., that they have the same meaning and that the data item can be used to access values for the attribute use class.

Attribute Use Class/Internal Schema Mapping (E250, AUC-IS Mapping)

Indicates that an attribute use class corresponds to some portion of an internal schema.

Attribute Use Class/Record Set Mapping (E135, AUC/Set Mapping)

Certain attribute use classes can be represented in a database by a group of record sets rather than by a data field. For example, Project Task Status might be represented by four Project:Task record sets called Pending, In-Process, On-Hold, and Completed. An attribute use class/record set mapping indicates that a particular record set corresponds to a particular attribute use class value.

Component Data Field (E195)

A data field that is part of another data field; e.g., if data field A is made up of data fields B, C, and D, each of these latter data fields is a component of A. A data field cannot be a component of more than one other data field.

Component Domain (E240)

An elementary domain that is part of another domain; e.g., a Date domain might be made up of a Month domain, a Day of Month domain, and a Year domain. Each of these latter domains would be a component of the Date domain. An elementary domain can be a component of several other domains.

Component Unit of Measure (E225)

An elementary unit of measure that is part of another unit of measure; e.g., the "inch" unit of measure is a component of the "foot-inch" unit of measure. An elementary unit of measure can be a component of several other units of measure.

Conceptual Schema/Internal Schema Complex Mapping Algorithm (E252, CS-IS Complex Mapping Algorithm)

A software module that must be used to access or transform data that is stored in a manner that the CDMP is not designed to handle.

Conceptual Schema/Internal Schema Mapping Parameter (E253, CS-IS Parameter)

A module parameter that is used to provide data to or get data from a CS-IS complex mapping algorithm.

Conceptual Schema/Internal Schema Mapping Parameter Constant (E255, CS-IS Parm Constant)

A static data value that is provided to a CS-IS complex mapping algorithm, via a parameter, to indicate what the algorithm is to do.

Constraint Attribute Use Class Reference (E219, Constraint AUC Reference)

Indicates that an attribute use class is mentioned somewhere within a constraint statement.

Constraint Entity Class Reference (E218, Constraint EC Reference)

Indicates that an entity class is mentioned somewhere within a constraint statement.

Constraint Input Parameter (E221)

A module parameter that is used to supply an attribute use class value to a generated constraint AP.

Constraint Module Reference (E217)

Indicates that a software module is mentioned in a constraint statement. A software module can be mentioned in the Failure Action clause of an assertion or in the Action clause of a trigger.

Constraint on Attribute Use Class Action (E216, Constraint on AUC Action)

An NDML action (SELECT or MODIFY) that is specified in conjunction with an attribute use class in the ON clause of a constraint statement.

Constraint on Entity Class Action (E215, Constraint on EC Action)

An NDML action (SELECT, INSERT, MODIFY, or DELETE) that is specified in conjunction with an entity class in the ON clause of a constraint statement.

Constraint Statement (E213)

One complete NDDL description of either an assertion, a trigger, or a horizontal partition fragment. An assertion is a rule about values for attribute use classes. If an NDML command attempts to violate an assertion, the CDMP rejects the command with an error message. A trigger is a set of conditions and a set of actions, both involving entity classes and attribute use classes. If the conditions are satisfied, all the actions are taken. If the conditions are not satisfied, none of the actions are taken. See the definitions of Horizontal Partition and Horizontal Partition Fragment for details about this use of constraint statements.

Constraint Statement Line (E214)

A fixed-length portion of a constraint statement.

Database Area (E102)

A subdivision of a CODASYL database. This subdivision is a technique for improving the efficiency accessing database record type instances. When a database is subdivided into database areas, some or all of its record types are assigned to particular areas. Instances of these record types are stored only within the assigned areas. Then, these record type instances can be accessed by searching only the appropriate areas rather than the entire database. This access method is only used when the record type instances cannot be located by other means (e.g., by calc keys or record sets).

Database Area Assignment (E103)

Indicates that a record type is assigned to a database area.

Database Directory (E259)

A software library that must be used when accessing a database.

Database Password (E25)

A code that must be supplied when logging on to a DBMS to use a database. The DBMS verifies the password before accepting any other messages.

Data Field (E67)

A portion of a record type in which data values can be stored.

Data Field Conceptual Schema/Internal Schema Mapping Parameter (E257, DF CS-IS Parameter)

A data field that is used as a CS-IS mapping parameter.

Data Field/Record Set Linkage (E41)

A data field in a variable data set in a TOTAL database that is used as the variable control key for a linkpath from a master data set.

Data Field Redefinition (E196)

A data field that occupies the same space in a record type as another data field. A record instance cannot contain values in both data fields. One instance can contain a value in one field while another contains a value in the other.

Data Format (E233)

The portion of a generic data description that includes the structural characteristics such as data type, length, storage method, etc. If a generic data description is for elementary values (e.g., customer names), it will have only one data format (e.g., Data Type - alphanumeric, Length - 30). If it is for compound values (e.g., part numbers consisting of six numerals followed by three letters followed by four more numerals), it will have more than one data format, one for each elementary

portion of the values. For the part number example the data
formats would be:

1. Data Type = numeric   Length = 6
2. Data Type = alphabetic   Length = 3
3. Data Type = numeric   Length = 4

A generic data description with a compound unit of measure,
i.e., one that is a group of component unit of measures, must
have a data format for each component unit of measure.

Data Item (E16)

An attribute class as seen by a user in a user view, i.e.,
a kind of data (e.g., employee hire date), not a particular data
value (e.g., 15 August 1980).

Data Management System (E23, DMS)

Either a database management system or a file management
system, i.e., a set of computer programs that must be used to
establish and maintain a database or a computer file.

Data Type (E93)

The combination of a type of values (e.g., alphanumeric,
signed numeric, etc.) and a type of storage (e.g., binary,
packed, etc.)

Description Type (E40)

A generic object may have several different kinds or styles
of description (short, long, technical, nontechnical, etc.).
Each is a description type.

DMS on Host (E20)

A data management system that is available on a particular
host.

Domain (E238)

A set of rules about the values that are allowed for a data item, attribute class, or data field. A domain is either an elementary domain of a group of two or more elementary domains, called component domains.

Domain Range (E242)

A series of consecutive values that represent all or part of an elementary domain.

Domain Value (E241)

A single value within an elementary domain.

Elementary Data Field (E194)

A data field that does not have any component data fields.

Elementary Domain (E239)

A domain that does not have any component domains. An elementary domain can be expressed as a series of values or value ranges.

Elementary Unit of Measure (E224)

A unit of measure that does not have any component units of measure.

Entity Class (E1)

A collection of similar entities, i.e., those that have the same kinds of attributes. An entity is a person, place, event, thing, concept, etc.

Entity Class (E1)

A relational join operation that combines two related

entity classes as part of the design of a record type.

Entity Class/Record Type Mapping (E204, EC-RT Mapping)

Indicates that an entity class corresponds to a record type, i.e., that they both have the same meaning and that the record type can be used to store instances of the entity class.

If a record type has more than one EC-RT mapping, some of its instances correspond to instances of one entity class while others correspond to instances of another, i.e., the record type is the relational union of the entity classes. An example is a Replenishment Order record type that maps to both the Purchase Order and Manufacturing Order entity classes. Each record instance represents either a purchase order or a manufacturing order.

Entity Class/Record Type Union Discriminator (E205, EC-RT Union Discriminator)

If a record type corresponds to more than one entity class, i.e., if it has more than one EC-RT mapping, it is the relational union of those entity classes. Some instances of such a record type correspond to instances of one of the entity classes, others to those of another. For such a record type there must be a way to determine which record instances correspond to instances of each entity class. An entity class/record type union discriminator provides this by specifying that a given value in a given data field indicates that a given EC-RT mapping should be used.

Entity Class/User View Join (E79, EC-UV Join)

A relational join operation that combines two related entity classes as part of the design of a user view.

File (E249)

A set of stored data that is managed by a file management system (e.g., VSAM).

File/Database (E24)

A set of stored data, i.e., either a computer file (e.g., a VSAM or flat file) or a database (e.g., an ORACLE or IMS database).

File/Database Reference (E249)

A computer file or database that is either accessed directly by a software module or is specified as one that the CDM Precompiler must use for MDML requests from that software module.

Generated Constraint Application Process (E220, Generated Constraint AP)

A software module that is created by the CDMP from a constraint statement. A generated constraint AP is used to evaluate the conditions specified in a constraint statement, to return a true-or-false response, and to carry out the specified actions, if any.

Generated Request Processor (E9, Generated RP)

A software module that was created by the CDMP Precompiler.

Generic Data Description (E232)

A detailed description of the values for one or more data items, attribute classes, data fields, and/or module parameters. It includes format, measurement, and domain characteristics of the values.

Generic Data Description Component Unit of Measure (E237, Gen Data Desc Component UOM)

A component unit of measure that is specified as part of a data format. These are only specified for a generic data description that includes a compound unit of measure, i.e., one that is a group of component units of measure.

Generic Data Description Domain (E244)

A domain that is specified as part of a generic data description.

Generic Data Description Unit of Measure (E236, Generic Data Description UOM)

A unit of measure that is specified as part of a generic data description.

Generic Object (E172)

Anything with a name that distinguishes it from other things of the same type and with a description that explains what it is (e.g., any entity class or attribute class).

Generic Object Description (E174)

An explanation of what a particular object is.

Generic Object Description Line (E175)

One fixed-length portion of a generic object description.

Generic Object Keyword (E173)

A keyword for a particular generic object.

Generic Object Name (E171)

An noun or noun phrase by which a generic object is known. Two objects can have the same name.

Horizontal Partition (E251)

Indicates that the same record type is not used to store all instances of an entity class, i.e., that one is used to store some instances while another is used to store others. Each record type represents a "fragment" of the entity class.

These fragments do not overlap, i.e., no entity instance appears
in more than one fragment. An entity class can be partitioned
into any number of fragments, usually with each being in a
different database or file, although that is not a requirement;
some or all may be stored as different record types in the same
database or file. A constraint statement defines each fragment,
i.e., describes the conditions that must be met by each entity
instance that is stored as a given record type. If an entity
class is replicated, i.e., if each of its instances is stored in
more than one database or file, each replication can be
horizontally partitioned. For example, for the first
replication the instances could be partitioned based on the
values in one attribute use class, and for the second
replication they could be partitioned based on the values in
another.

Horizontal Partition Fragment (E212)

A record type that is used to store some, but not all, of
the instances of an entity class. A constraint statement
describes the conditions that must be met by each entity
instance that is stored as the record type. If the conditions
are satisfied by the attribute values of an entity instance, it
can be stored as an instance of the record type; otherwise, it
cannot be.

Host (E21)

A computer in the IISS.

IMS Segment (E81)

A record type in a database that is controlled by IBM's IMS
DBMS.

Inherited Attribute Class (E7)

A key class member that has migrated from the independent
entity class of a relation class to become an attribute use
class in the dependent entity class.

Inherited Key Class (E143)

A key class in the independent entity class of a relation class that has migrated to appear in the dependent entity class of that relation class.

Key Class (E3)

A group of one or more of an entity's attributes that can be used to uniquely identify the entity within its entity class. An entity can have more than one key. A key class is a collection of the attribute classes whose member attributes comprise the keys for the entities in an entity class. An entity class has the same number of key classes as each of its member entities has keys. For example, if each entity has three keys, the entity class has three key classes.

Key Class Member (E6)

An attribute use class that is part of a key class.

Library Module (E248)

A software module that is stored in a software library.

Model (E155)

A representation of the information requirements of all or part of an enterprise in terms of entity classes, relation classes, and attribute classes. Model Glossary Name (E176)

A name of a model entity class or a model attribute class, either an official name or an alias.

Module Parameter (E59)

A means of supplying values to a software module and of receiving results from a module.

Numeric Data Format (E235)

A data format for values that can only contain numerals (0-9) and associated punctuation (decimal point, comma, etc.).

Owned Attribute Class (E140)

A model attribute class that appears as an attribute use class in a model entity class and is not an inherited attribute class.

Program Control Block (E78, PCB)

A portion of a PSB that describes and controls how an IMS database can be accessed.

Program Specification Block (E77, PSB)

A group of logical views of IMS databases that is used for interacting with the IMS DBMS.

Record Set (E72)

An association between one record type, called the owner, and one or more other record types, called the members.

Record Set Member (E134)

A record type that is a member of a record set.

Record Type (E66)

A group of data values that are stored together as a unit in a computer file or database. A record type is the collection of all the records of the same kind, i.e., all the records that contain the same kind of data values.

Record Type Conceptual Schema/Internal Schema Mapping Parameter (E256, RT CS-IS Parameter)

A record type that is used as a CS-IS mapping parameter.

Relation Class (E4)

An association between an entity in one entity class and one in another. A relationship has a label that describes the association. For example, a customer named ABC Corp. is associated with an order numbered 123 in a manner labeled "placed". A elation class is a collection of the identically labeled relationships between the members of the same two entity classes. Each relation class is either specific or nonspecific.

In a specific relation class one entity class is "independent" while the other is "dependent"; i.e., entities in the first can exist without being associated with any in the second, but those in the second cannot exist without being associated with one in the first. One key class from the independent entity class "migrates" through each specific relation class to appear in the dependent entity class as inherited attribute classes.

In a nonspecific relation class, neither entity class is dependent on the other; i.e., entities in either entity class can exist without being associated with any in the other. For convenience, one entity class is arbitrarily called "independent" and the other is called "dependent".

Relation Class/Record Set Mapping (E109, RC-Set Mapping)

Indicates that a record set represents the same association as a relation class. If a record set has more than one member record type, it may represent several relation classes, a different one for each member. Hence, this entity class is only indirectly dependent on record set (via record set member).

Repeating Data Field Occurrence Counter (E258, Repeating DF Occurrence Counter)

A data field whose data values indicate how many occurrences of a repeating data field actually contain values.

Segment Data Field (E82)

A data field in an IMS segment.

Software Library (E247)

A computer file in which software modules can be stored.

Software Module (E57)

A set of computer instructions that are treated as a whole (i.e., stored, compiled, and executed together).

Subschema (E14)

The description, in the DDL of a CODASYL DBMS, of all or part of a database. For IISS only one subschema is needed for a CODASYL database, and it must describe all the common data within the database that is to be accessible with NDML.

Unit of Measure (E223)

A standard scale for determining the magnitude of something. Examples include inch, foot, foot-inch, meter, ounce, pound, hour, minute, second, etc.

Unit of Measure Conversion (E226)

A means of transforming a value expressed in one unit of measure into an equivalent value expressed in another (e.g., transforming inches to feet or feet to meters).

Unit of Measure Conversion Constant (E230, UOM Conversion Constant)

A number in a unit of measure conversion step that is the same every time the conversion is performed.

Unit of Measure Conversion Module (E227, UOM Conversion Module)

A software module that can be used to perform a unit of measure conversion.

Unit of Measure Conversion Parameter (E228, UOM Conversion Parameter)

A module parameter that is used to supply values to or receive values from a unit of measure conversion module.

Unit of Measure Conversion Step (E229, UOM Conversion Step)

One of a series of arithmetic steps that can be used to perform a unit of measure conversion. Each step takes the value resulting from the prior step (the first step uses the value to be converted) and adds, subtracts, multiplies, or divides by another value, either a constant or a variable. The result of the last step is the converted value. The processing sequence is always first step to last; parentheses, branching, and conditional tests are not allowed. Consequently, some unit of measure conversions cannot be performed in this manner (e.g., converting meters to feet-and-inches).

Unit of Measure Conversion Variable (E231, UOM Conversion Variable)

A number in a unit of measure conversion step that can be different every time the conversion is performed. This is only used when the unit of measure being converted from has two or more component units of measure. Each component is a variable and each is involved in a separate step.

User Application Process (E8, User AP)

A software module that supports business activities rather than data processing activities and that can be executed directly, i.e., a main routine, not a subroutine. A user AP may contain NDML commands for accessing stored data via the CDM, or it may access them directly via DMSs, or it may call subroutines that contain NDML commands or that access stored data directly.

User View (E15)

A group of data items that a user wants to deal with as a group. It is similar to an entity class but does not necessarily meet all the conditions for being one; it can be thought of as an unnormalized entity class. A user view is often the result of combining several entity classes via relational join operations and selecting particular attribute use classes as data items via relational project operations.

User View Reference (E116)

A user view against which a software module poses NDML commands.

Verification Module (E243)

A software module that can be used to determine whether a value conforms to a domain.

4.2 Owned and Inherited Attribute Classes

Entity Class/Attribute Class Name	Source	Key
Alpha-Numeric Data Format:		
Data Description Object Number	E233	K1
Format Number	E233	K1
Maximum Length	Owned	
Fill Character	Owned	
Alphabetic Case Code	Owned	
Attribute Class:		
Model Object Number	E176	K1,K2
Attribute Class Object Number	E176	K1
Attribute Class Label	Owned	K2

Attribute Class Data Description:

Model Object Number	E2	K1
Attribute Class Object Number	E2	K1
Data Description Object Number	E232	

Attribute Use Class:

Model Object Number	E1,E140,E171	K1,K2,K3
Attribute Use Class Number	Owned	K1
Entity Class Object Number	E1	K2,K3
Attribute Use Class Object Name	E171	K2
Attribute Use Class Label	Owned	K3
Attribute Class Object Number	E140	

Attribute Use Class Conceptual Schema/Internal Schema Mapping Parameter:

Model Object Number	E253,E250	K1,K2
Entity Class Object Number	E253,E250	K1,K2
File/Database Object Number	E253,E250	K1,K2
Record Type Object Name	E253,E250	K1,K2
Module Identification	E253	K1
Parameter Sequence Number	E253	K1
Attribute Use Class Number	E250	K2

Attribute Use Class/Data Field Mapping:

Model Object Number	E250	K1
Entity Class Object Number	E250	K1
File/Database Object Number	E250	K1
Record Type Object Name	E250	K1
Attribute Use Class Number	E250	K1
Data Field Object Name	E67	
Repeating Data Field Index Indicator	Owned	

Attribute Use Class/Data Item Mapping:

User View Object Number	E16	K1
Data Item Object Name	E16	K1
Model Object Number	E5	
Attribute Use Class Number	E5	

Attribute Use Class/Internal Schema Mapping:

Model Object Number	E204,E5	K1
Entity Class Object Number	E204	K1
File/Database Object Number	E204	K1
Record Type Object Name	E204	K1
Attribute Use Class Number	E5	K1
Preference Number	Owned	

Attribute Use Class/Record Set Mapping:

Model Object Number	E250	K1
Entity Class Object Number	E250	K1
File/Database Object Number	E250,E72	K1,K2
Record Type Object Name	E250	K1
Attribute Use Class Number	E250	K1
Record Set Object Name	E72	K2
Equivalent Attribute Use Class Value	Owned	

Component Data Field:

File/Database Object Number	E67,E67	K1
Record Type Object Name	E67,E67	K1
Component Data Field Object Name	E67	K1
Group Data Field Object Name	E67	

Component Domain:

Group Domain Object Number	E238	K1,K2
Elementary Domain Object Number	E239	K1
Component Domain Sequence Number	Owned	K2

Component Unit of Measure:

Group UOM Code	E223	K1,K2
Elementary UOM Code	E224	K1
Component UOM Sequence Number	Owned	K2

Conceptual Schema/Internal Schema Complex Mapping Algorithm:

Model Object Number	E204	K1

Entity Class Object Number	E204	K1
File/Database Object Number	E204	K1
Record Type Object Name	E204	K1
Module Identification	E57	K1
CS-IS Mapping Algorithm Use Code	Owned	

Conceptual Schema/Internal Schema Mapping Parameter:

Model Object Number	E252	K1
Entity Class Object Number	E252	K1
File/Database Object Number	E252	K1
Record Type Object Name	E252	K1
Module Identification	E252,E59	K1
Parameter Sequence Number	E59	K1

Conceptual Schema/Internal Schema Mapping Parameter Constant:

Model Object Number	E253	K1
Entity Class Object Number	E253	K1
File/Database Object Number	E253	K1
Record Type Object Name	E253	K1
Module Identification	E253	K1
Parameter Sequence Number	E253	K1
CS-IS Mapping Constant Use Code	Owned	K1
CS-IS Mapping Constant Value	Owned	

Constraint Attribute Use Class Reference:

Statement Object Number	E213	K1
Model Object Number	E5	K1
Attribute Use Class Number	E5	K1

Constraint Entity Class Reference:

Statement Object Number	E213	K1
Model Object Number	E1	K1
Entity Class Object Number	E1	K1

Constraint Input Parameter:

Module Identification	E59,E220	K1
Parameter Sequence Number	E59	K1

Statement Object Number	E219,E220	K2
Model Object Number	E219	K2
Attribute Use Class Number	E219	K2

Constraint Module Reference:

Statement Object Number	E213	K1
Module Identification	E57	K1

Constraint on Attribute Use Class Action:

Statement Object Number	E219,E215	K1
Model Object Number	E219,E215	K1
Attribute Use Class Number	E219	K1
Entity Class Object Number	E215	
Statement Action	E215	K1

Constraint on Entity Class Action:

Statement Object Number	E218,E213	K1
Model Object Number	E218	K1
Entity Class Object Number	E218	K1
Statement Action	Owned	K1

Constraint Statement:

Statement Object Number	E171	K1
Statement Object Name	E171	
Statement Type Code	Owned	

Constraint Statement Line:

Statement Object Number	E213	K1
Statement Line Number	Owned	K1
Statement Line Text	Owned	

Database Area:

File/Database Object Number	E24	K1
Database Area Identification	Owned	K1

Database Area Assignment:

File/Database Object Number	E102,E66	K1
Database Area Identification	E102	K1
Record Type Object Name	E66	K1

Database Directory:

File/Database Object Number	E24	K1
Host Identification	E247	
Library Identification	E247	

Database Password:

File/Database Object Number	E24	K1
Database Password	Owned	

Data Field:

File/Database Object Number	E66	K1,K2
Record Type Object Name	E66	K1,K2
Data Field Object Number	E171	
Data Field Object Name	E171	K1
Data Field Sequence Number	Owned	K2
Record Key Code	Owned	
Number of Data Field Occurrences	Owned	
DBMS-Accessible Data Field Indicator	Owned	

Data Field Conceptual Schema/Internal Schema Mapping Parameter:

Model Object Number	E253	K1
Entity Class Object Number	E253	K1
File/Database Object Number	E253,E67	K1
Record Type Object Name	E253,E67	K1
Module Identification	E253	K1
Parameter Sequence Number	E253	K1
Data Field Object Name	E67	

Data Field/Record Set Linkage:

File/Database Object Number	E67,E134	K1,K2

Record Type Object Name	E67,E134	K1,K2
Data Field Object Name	E67	K1
Record Set Object Name	E134	K2
Linkage Type Code	Owned	

Data Field Redefinition:

File/Database Object Number	E67,E67	K1
Record Type Object Name	E67,E67	K1
Redefining Data Field Object Name	E67	K1
Redefined Data Field Object Name	E67	

Data Format:

Data Description Object Number	E232	K1
Format Number	Owned	K1
Data Type Code	E93	

Data Item:

User View Object Number	E15	K1
Data Item Object Number	E171	
Data Item Object Name	E171	K1

Data Management System:

DMS Name	Owned	K1
DMS Type Code	Owned	

Data Type:

Data Type Code	Owned	K1
Data Type Name	Owned	K2

Description Type:

Description Type Code	Owned	K1
Description Type Name	Owned DMS on Host:	
DMS Name	E23	K1
Host Identification	E21	K1

Null Value   Owned

Domain:

    Domain Object Number   E171   K1
    Domain Object Name   E171

Domain Range:

    Elementary Domain Object Number   E239   K1
    Beginning Value   Owned   K1
    Ending Value   Owned   K1

Domain Value:

    Elementary Domain Object Number   E239   K1
    Specific Value   Owned   K1

Elementary Data Field:

    File/Database Object Number   E67   K1
    Record Type Object Name   E67   K1
    Data Field Object Name   E67   K1
    Data Description Object Number   E232

Elementary Domain:

    Elementary Domain Object Number   E238   K1
    Data Description Object Number   E232

Elementary Unit of Measure:

    Elementary UOM Code   E223   K1

Entity Class:

    Model Object Number   E176   K1,K2
    Entity Class Object Number   E176   K1
    Entity Class Label   Owned   K2

4-25

Entity Class/Record Type Join:

Model Object Number	E204,E143	K1
Entity Class Object Number	E204	K1
File/Database Object Number	E204	K1
Record Type Object Name	E204	K1
Relation Class Object Number	E143	K1
EC-RT Join Type Code	Owned	

Entity Class/Record Type Mapping:

Model Object Number	E1	K1
Entity Class Object Number	E1	K1
File/Database Object Number	E66	K1
Record Type Object Name	E66	K1

Entity Class/Record Type Union Discriminator:

Model Object Number	E204	K1
Entity Class Object Number	E204	K1
File/Database Object Number	E204,E67	K1
Record Type Object Name	E204,E67	K1
Data Field Object Name	E67	K1
EC-RT Union Value	Owned	

Entity Class/User View Join:

Model Object Number	E143	K1
Relation Class Object Number	E146	K1
User View Object Number	E15	K1
EC-UV Join Type Code	Owned	

File:

File/Database Object Number	E24	K1
Blocking Factor	Owned	
Device Type Code	Owned	

File/Database:

File/Database Object Number	E171	K1

File/Database Object Name	E171	K2
DMS Name	E20	
Host Identification	E20	

File/Database Reference:

Module Identification	E57	K1
File/Database Object Number	E24	K1
File/Database Reference Type Code	Owned	

Generated Constraint Application Process:

Module Identification	E57	K1
Statement Object Number	E213	K2

Generated Request Processor:

Generated Module Identification	E57	K1
Generating Module Identification	E57	

Generic Data Description:

Data Description Object Number	E171	K1
Data Description Object Name	E171	

Generic Data Description Component Unit of Measure:

Data Description Object Number	E235,E236	K1
Format Number	E235	K1
Group UOM Code	E225	
Elementary UOM Code	E225	

Generic Data Description Domain:

Data Description Object Number	E232	K1
Domain Object Number	E238	

Generic Data Description Unit of Measure:

Data Description Object Number	E232	K1
UOM Code	E223	

Generic Object:

Object Number	Owned	K1
Object Creation Date	Owned	
Object Modification Date	Owned	

Generic Object Description:

Object Number	E172	K1
Description Type Code	E40	K1
Author Identification	Owned	

Generic Object Description Line:

Object Number	E174	K1
Description Type Code	E174	K1
Description Line Number	Owned	K1
Description Text	Owned	

Generic Object Keyword:

Object Number	E172	K1
Keyword	Owned	K1

Generic Object Name:

Object Number	E172	K1
Object Name	Owned	K1
Object Name Type Code	Owned	

Horizontal Partition:

Model Object Number	E1	K1
Entity Class Object Number	E1	K1
Horizontal Partition Number	Owned	

Horizontal Partition Fragment:

Model Object Number	E251	K1

Entity Class Object Number	E251	K1
Horizontal Partition Number	E251	
File/Database Object Number	E204	K1
Record Type Object Name	E204	K1

Host:

Host Identification	Owned	K1

IMS Segment:

File/Database Object Number	E66	K1
Record Type Object Name	E66	K1
Segment Size	Owned	

Inherited Attribute Class:

Model Object Number	E5,E143,E6	K1
Attribute Use Class Number	E5	K1
Relation Class Object Number	E143	
Key Class Number	E6	
KC Member Attribute Use Class Number	E6	

Inherited Key Class:

Model Object Number	E4,E3	K1
Relation Class Object Number	E4	K1
Key Class Number	E3	

Key Class:

Model Object Number	E1	K1
Entity Class Object Number	E1	
Key Class Number	Owned	K1

Key Class Member:

Model Object Number	E3,E5	K1
Key Class Number	E3	K1
Attribute Use Class Number	E5	K1

Library Module:

Host Identification	E247	K1
Library Identification	E247	K1
Module Identification	E57	K1

Model:

Model Object Number	E171	K1
Model Object Name	E171	K2

Model Glossary Name:

Model Object Number	E155	K1,K2
Object Number	E171	K1
Object Name	E171	K2

Module Parameter:

Module Identification	E57	K1,K2
Parameter Identification	Owned	K1
Parameter Sequence Number	Owned	K2
Data Description Object Number	E232	
Parameter Use Code	Owned	

Numeric Data Format:

Data Description Object Number	E233	K1
Format Number	E233	K1
Integer Length	Owned	
Decimal Length	Owned	
Round/Truncate Code	Owned	
Storage Type Code	Owned	

Owned Attribute Class:

Model Object Number	E2,E1	K1
Attribute Class Object Number	E2	K1
Entity Class Object Number	E1	

Program Control Block:

File/Database Object Number	E24	K1
PSB Name	E77	
PSB Sequence Number	Owned	
Key Feedback Length	Owned	

Program Specification Block:

PSB Name	Owned	K1
Host Identification	E21	

Record Set:

File/Database Object Number	E66	K1
Record Set Object Name	E171	K1
Record Set Object Number	E171	
Record Type Object Name	E66	
Total Number of Members	Owned	

Record Set Member:

File/Database Object Number	E72,E66	K1
Record Set Object Name	E72	K1
Record Type Object Name	E66	K1
Required Membership Indicator	Owned	

Record Type:

File/Database Object Number	E24	K1
Record Type Object Name	E171	K1
Record Type Object Number	E171	

Record Type Conceptual Schema/Internal Schema Mapping Parameter:

Model Object Number	E253	K1
Entity Class Object Number	E253	K1
File/Database Object Number	E253	K1
Record Type Object Name	E253	K1
Module Identification	E253	K1
Parameter Sequence Number	E253	K1

Relation Class:

Model Object Number	E1,E1	K1,K2
Relation Class Object Number	E171	K1
Independent Entity Class Object Number	E1	K2
Dependent Entity Class Object Number	E1	K2
Relation Class Object Name	E171	K2
Minimum Number of Independent Entities	Owned	
Maximum Number of Independent Entities	Owned	
Minimum Number of Dependent Entities	Owned	
Maximum Number of Dependent Entities	Owned	

Relation Class/Record Set Mapping:

Model Object Number	E143	K1
Relation Class Object Number	E143	K1
File/Database Object Number	E134	K1
Record Set Object Name	E134	K1
Record Type Object Name	E134	K1

Repeating Data Field Occurrence Counter:

File/Database Object Number	E67,E67	K1,K2
Record Type Object Name	E67,E67	K1,K2
Counter Data Field Object Name	E67	K1
Repeating Data Field Object Name	E67	K2

Segment Data Field:

File/Database Object Number	E67	K1
Record Type Object Name	E67	K1
Data Field Object Name	E67	K1
Segment Starting Byte	Owned	

Software Library:

Host Identification	E21	K1
Library Identification	Owned	K1
Library Type Code	Owned	

Software Module:

Module Identification	Owned	K1
Object Number	E171	K2
Object Name	E171	K2
Software Language Name	Owned	
Latest Usage Date	Owned	

Subschema:

File/Database Object Number	E24	K1
Schema Name	Owned	
Subschema Name	Owned	

Unit of Measure:

UOM Code	Owned	K1
UOM Name	Owned	K2

Unit of Measure Conversion:

From UOM Code	E223	K1
To UOM Code	E223	K1

Unit of Measure Conversion Constant:

From UOM Code	E229	K1
To UOM Code	E229	K1
UOM Step Number	E229	K1
UOM Constant Value	Owned	

Unit of Measure Conversion Module:

From UOM Code	E226	K1
To UOM Code	E226	K1
Module Identification	E57	

Unit of Measure Conversion Parameter:

From UOM Code	E227	K1
To UOM Code	E227	K1

Elementary UOM Code	E224	K1
Module Identification	E59	
Parameter Sequence Number	E59	

Unit of Measure Conversion Step:

From UOM Code	E226	K1
To UOM Code	E226	K1
UOM Step Number	Owned	K1
UOM Step Operator	Owned	

Unit of Measure Conversion Variable:

From UOM Code	E229	K1
To UOM Code	E229	K1
UOM Step Number	E229	K1
Elementary UOM Code	E224	

User Application Process:

User AP Module Identification	E57	K1

User View:

User View Object Number	E171	K1
User View Object Name	E171	K2

User View Reference:

Module Identification	E57	K1
User View Object Name	E15	K1

Verification Module:

Domain Object Number	E238	K1
Module Identification	E57	

4.5 CDM1 Entity Class Glossary Index by Entity Class Number

EC No	Entity Class Name
1	Entity Class
2	Attribute Class
3	Key Class
4	Relation Class
5	Attribute Use Class
6	Key Class Member
7	Inherited Attribute Class
8	User Request Processor
9	Generated Application Process
14	Subschema
15	User View
16	Data Item
20	DMS on Host
21	Host
23	Data Management System
24	File/Database
25	Database Password
40	Description Type
41	Data Field/Record Set Linkage
57	Software Module
59	Module Parameter
64	Attribute Use Class/Data Item Mapping
66	Record Type
67	Data Field
72	Record Set
76	Entity Class/Record Type Join
77	Program Specification Block
78	Program Control Block
79	Entity Class/User View Join
81	IMS Segment
82	Segment Data Field
93	Data Type
102	Database Area
103	Database Area Assignment
108	Attribute Use Class/Data Field Mapping
109	Relation Class/Record Set Mapping
116	User View Reference
134	Record Set Member
135	Attribute Use Class/Record Set Mapping
140	Owned Attribute Class
143	Inherited Key Class
155	Model

EC No	Entity Class Name
171	Generic Object Name
172	Generic Object
173	Generic Object Keyword
174	Generic Object Description
175	Generic Object Description Line
176	Model Glossary Name
184	Attribute Class Data Description
194	Elementary Data Field
195	Component Data Field
196	Data Field Redefinition
204	Entity Class/Record Type Mapping
205	Entity Class/Record Type Union Discriminator
212	Horizontal Partition Fragment
213	Constraint Statement
214	Constraint Statement Line
215	Constraint On Entity Class Action
216	Constraint On Attribute Use Class Action
217	Constraint Module Reference
218	Constraint Entity Class Reference
219	Constraint Attribute Use Class Reference
220	Generated Constraint Application Process
221	Constraint Input Parameter
223	Unit of Measure
224	Elementary Unit of Measure
225	Component Unit of Measure
226	Unit of Measure Conversion
227	Unit of Measure Conversion Module
228	Unit of Measure Conversion Parameter
229	Unit of Measure Conversion Step
230	Unit of Measure Conversion Constant
231	Unit of Measure Conversion Variable
232	Generic Data Description
233	Data Format
234	Alpha-Numeric Data Format
235	Numeric Data Format
236	Generic Data Description Unit of Measure
237	Generic Data Description Component Unit of Measure
238	Domain
239	Elementary Domain
240	Component Domain
241	Domain Value
242	Domain Range
243	Verification Module
244	Generic Data Description Domain
247	Software Library

EC No	Entity Class Name
248	Library Module
249	File/Database Reference
250	Attribute Use Class/Internal Schema Mapping
251	Horizontal Partition
252	Conceptual Schema/Internal Schema Complex Mapping Algorithm
253	Conceptual Schema/Internal Schema Mapping Parameter
254	Attribute Use Class Conceptual Schema/Internal Schema Mapping Parameter
255	Conceptual Schema/Internal Schema Mapping Parameter Constant
256	Record Type Conceptual Schema/Internal Schema Mapping Parameter
257	Data Field Conceptual Schema/Internal Schema Mapping Parameter
258	Repeating Data Field Occurrence Counter
259	Database Directory

4.4 CDM1 Entity Class Glossary Index by Entity Class Label

Note:
> Only entity classes whose labels are different than their names are listed here. If an entity class label is not included, it is the same as the entity class name.

Entity Class Label	Entity Class Name
AUC CS-IS Parameter	Attribute Use Class Conceptual Schema/ Internal Schema Mapping Parameter
AUC-DF Mapping	Attribute Use Class/Data Field Mapping
AUC-ι Mapping	Attribute Use Class/Data Item Mapping
AUC-IS Mapping	Attribute Use Class/Internal Schema Mapping
AUC-Set Mapping	Attribute Use Class/Record Set Mapping
Constraint AUC Reference	Constraint Attribute Use Class Reference
Constraint EC Reference	Constraint Entity Class Reference
Constraint On AUC Action	Constraint On Attribute Use Class Action
Constraint On EC Action	Constraint On Entity Class Action

Entity Class Label	Entity Class Name
CS-IS Complex Mapping Algorithm	Conceptual Schema/Internal Schema Complex Mapping Algorithm
CS-IS Parameter	Conceptual Schema/Internal Schema Mapping Parameter
CS-IS Parm Constant	Conceptual Schema/Internal Schema Mapping Parameter Constant
DF CS-IS Parameter	Data Field Conceptual Schema/Internal Schema Mapping Parameter
DMS	Data Management System
EC-RT Join	Entity Class/Record Type Join
EC-RT Mapping	Entity Class/Record Type Mapping
EC-RT Union Discriminator	Entity Class/Record Type Union Discriminator
EC-UV Join	Entity Class/User View Join
Gen Data Desc Component UOM	Generic Data Description Component Unit of Measure
Generated Constraint AP	Generated Constraint Application Process
Generated RP	Generated Request Processor
Generic Data Description UOM	Generic Data Description Unit of Measure
Horiz Partition Fragment	Horizontal Partition Fragment
PCB	Program Control Block
PSB	Program Specification Block
RC-Set Mapping	Relation Class/Record Set Mapping
Repeating DF Occurrence Counter	Repeating Data Field Occurrence Counter
RT CS-IS Parameter	Record Type Conceptual Schema/Internal Schema Mapping Parameter
UOM Conversion Constant	Unit of Measure Conversion Constant
UOM Conversion Module	Unit of Measure Conversion Module
UOM Conversion Parameter	Unit of Measure Conversion Parameter
UOM Conversion Step	Unit of Measure Conversion Step
UOM Conversion Variable	Unit of Measure Conversion Variable
User AP	User Application Process

SECTION 5

ATTRIBUTE CLASS DOCUMENTATION

5.1 CDM1 Attribute Class Glossary

Note:

The attribute classes in this glossary are in alphabetic sequence by attribute class name. Attribute class numbers appear in parentheses following the names. Attribute class labels that are differnet than the names appear with the numbers. If a label is the same as a name, the label is omitted.

Role names for inherited attribute classes are also included in this glossary. Role names are variations of attribute class names that are used to distinguish between inherited attribute classes in an entity class that migrated from the same key class member via different relation classes. Attribute class numbers are not shown with these glossary entries; inherited attribute classes have the same attribute class numbers as the corresponding key class members.

Alphabetic Case Code (A160, Alpha Case Code)

A code in an alpha-numeric data format that indicates whether only upper case letters, only lower case letters, or both are permitted.

Attribute Class Label (A161, AC Label)

An abbreviated version of an attribute class object name (or the entire name if it is fairly short). It is used to uniquely identify an attribute class in model diagrams.

Attribute Class Object Number (AC Obj No)

Role name for Object Number; introduced in the Attribute Class entity class.

Attribute Use Class Label (A14, AUC Label)

An abbreviated version of an attribute use class object
name (or the entire name if it is fairly short) that
distinguishes an attribute use class from the others in the
same model entity class.

Attribute Use Class Number (A12, AUC No)

An identification code that distinguishes an attribute use
class from the others in the same model.

Attribute Use Class Object Name (AUC Obj Name)

Role name for Object name; introduced in the Attribute Use
Class entity class.

Author Identification (A110, Author Id)

The means of identifying who wrote a generic object
description.

Beginning Value (A162, Begin Value)

The lowest value in a domain range.

Blocking Factor (A193)

The number of records that can be physically stored
together in one block in a file.

Component Data Field Object Name (Comp DF Obj Name)

Role name for Data Field Object Name; introduced in the
Component Data Field entity class.

Component Domain Sequence Number (A164, Comp Dom Seq No)

The ordinal position of a component domain within the group
of those for a domain.

Component UOM Sequence Number (A165, Comp UOM Seq No)

The ordinal position of a component unit of measure within the group of those for a unit of measure.

Counter Data Field Object Name (Ctr DF Obj Name)

Role name for Data Field Object Name; introduced in the Repeating Data Field Occurrence Counter entity class.

CS-IS Mapping Algorithm Use Code (A198, CS-IS Alg Use Code)

A code that indicates whether a CS-IS complex mapping algorithm can be used on retrievals only, on updates only, or on both.

CS-IS Mapping Constant Use Code (A199, CS-IS Constant Use Code)

A code that indicates whether a CS-IS mapping constant is to be used for retrievals or updates.

CS-IS Mapping Constant Value (A200, CS-IS Constant Value)

A static data value that is provided to a CS-IS complex mapping algorithm via a parameter.

Database Area Identification (A75, Area Id)

A code that distinguishes a database area from the others in the same database.

Database Password (A60, DB Password)

A code that must be supplied when logging on to a DBMS to use a database. The DBMS verifies the password before accepting any other messages.

Data Description Object Name (DD Obj Name)

Role name for Object Name; introduced in the Data
Description entity class.

Data Description Object Number (DD Obj No)

Role name for Object Number; introduced in the Data
Description entity class.

Data Field Object Name (DF Obj Name)

Role name for Object Name; introduced in the Data Field
entity class.

Data Field Object Number (DF Obj No)

Role name for Object Number; introduced in the Data Field
entity class.

Data Field Sequence Number (A163, DF Seq No)

The ordinal position of a data field within a record type.

Data Item Object Name (DI Obj Name)

Role name for Object Name; introduced in the Data Item
entity class.

Data Item Object Number (DI Obj No)

Role name for Object Number; introduced in the Data Item
entity class.

Data Type Code (A76, DT Code)

A code that indicates what kind of values can be
represented by a data type. The types are alphabetic,
alphanumeric, signed numeric, and unsigned numeric.

Data Type Name (A166, DT Name)

A noun or noun phrase that briefly describes and uniquely
identifies a data type.

DBMS-Accessible Data Field Indicator (A122, DBMS-Acc DF Ind)

A code that indicates whether a data field in a database is
"visible to or hidden from" the DBMS, i.e., whether the
data field can be explicitly included in a DBMS
transaction.

Decimal Length (A167, Dec Len)

The greatest number of numerals that can be held in the
decimal portion of a numeric data format.

Dependent Entity Class Object Number (Dep EC Obj No)

Role name for Entity Class Object Number; introduced in the
Relation Class entity class.

Description Line Number (A116, Desc Line No)

A number that distinguishes a generic object description
line from the others in the same generic object
description.

Description Text (A117, Desc Text)

A fixed-length portion of a generic object description.

Description Type Code (A118, Desc Type Code)

A code that uniquely identifies a description type.

Description Type Name (A168, Desc Type Name)

A noun or noun phrase that briefly describes a description
type.

Device Type Code (A194)

A code that identifies the type of data storage device that a file resides on.

DMS Name (A58)

The name by which a data management system is commonly known. For example, IMS, ORACLE, TOTAL, and IDMS are DBMSs; VSAM is a a file management system.

DMS Type Code (A59)

Indicates whether a data management system is for files or databases and, if it is for databases, whether they are relational, hierarchical, or network databases.

Domain Object Name (Dom Obj Name)

Role name for Object Name; introduced in the Domain entity class.

Domain Object Number (Dom Obj No)

Role name for Object Number; introduced in the Domain entity class.

EC-RT Join Type Code (A115)

A code that indicates how to join two entity classes in creating a record type (e.g., natural join or outer join).

EC-RT Union Discriminator Value (A169, EC-RT Union Discrim Value)

A value that exists in a specified data field in every record instance for which an EC-RT mapping is valid. This only exists for record types that result from relational unions of entity classes.

EC-UV Join Type Code (A112)

A code that indicates how to join two entity classes in creating a user view (e.g., natural join or outer join).

Elementary Domain Object Number (Elem Dom Obj No)

Role name for Domain Object Number; introduced in the Elementary Domain entity class.

Elementary UOM Code (Elem UOM Code)

Role name for UOM Code; introduced in the Elementary Unit of Measure entity class.

Ending Value (A170, End Value)

The highest value in a domain range.

Entity Class Label (A3, EC Label)

An abbreviated version of an entity class object name (or the entire name if it is fairly short). It is used to uniquely identify an entity class in model diagrams.

Entity Class Object Number (EC Obj No)

Role name for Object Number; introduced in the Entity Class entity class.

Equivalent Attribute Use Class Value (A108, Equiv AUC Value)

A value for an attribute use class that is represented by a record set.

File/Database Object Name (F/DB Obj Name)

Role name for Object Name; introduced in the File/Database entity class.

File/Database Object Number (F/DB Obj No)

Role name for Object Number; introduced in the
File/Database entity class.

File/Database Reference Type Code (A195, F/DB Ref Type Code)

A code that indicates whether a computer file or database
that is referenced by a software module is accessed
directly by that module or is just identified as one that
the CDM Precompiler must use to respond to NDML requests
from that module.

Fill Character (A172, Fill Char)

A value that is put in unused positions of an alpha-
numeric data format when a value that is shorter than the
maximum length is moved to it.

Format Number (A174, Format No)

The ordinal position of a data format within all those for
a generic data description.

From UOM Code

Role name for UOM Code; introduced in the Unit of Measure
Conversion entity class.

Generated Module Identification (Generated Mod Id)

Role name for Module Identification; introduced in the
Generated Request Processor entity class.

Generating Module Identification (Generating Mod Id)

Role name for Module Identification; introduced in the
Generated Request Processor entity class.

Group Data Field Object Name (Group DF Obj Name)

Role name for Data Field Object Name; introduced in the Component Data Field entity class.

Group Domain Object Number (Group Dom Obj No)

Role name for Domain Object Number; introduced in the Component Domain entity class.

Group UOM Code

Role name for UOM Code; introduced in the Component Unit of Measure entity class.

Horizontal Partition Number (A201, HP No)

A number that distinguishes one horizontal partition from any others for an entity class.

Host Identification (A121, Host Id)

The means of uniquely identifying a host.

Independent Entity Class Object Number (Ind EC Obj No)

Role name for Entity Class Object Number; introduced in the Relation Class entity class.

Integer Length (A175, Int Len)

The greatest number of numerals that can be held in the integer portion of a numeric data format.

KC Member Attribute Use Class Number (KC Mbr AUC No)

Role name for Attribute Use Class Number; introduced in the Inherited Attribute Class entity class.

Key Class Number (A9, KC No)

An identification code that distinguishes a key class from
the others in the same model.

Key Feedback Length (A123, Key Feedback Len)

The maximum size of the concatenated keys from each seg-
ment in an IMS database, from the root to the bottom child
segment along any one branch.

Keyword (A124)

A word or phrase that has been designated as a means of
locating a generic object or a number of similar generic
objects.

Latest Usage Date (A30)

The date when a software module was last executed.

Library Identification (A176, Lib Id)

A code that distinguishes a software library from the
others on the same host.

Library Type Code (A177, Lib Type Code)

A code that indicates what kind of software modules are
stored in a software library (e.g., source modules, object
modules).

Linkage Type Code (A125, Linkage Type)

A code that indicates whether a data field that is used for
a set linkage is a symbolic calc key (S) or an actual
database key (K). In a TOTAL database these are all
symbolic calc keys.

Maximum Length (A178, Max Len)

The greatest number of letters, numerals, punctation, etc.,
that can be held in an alpha-numeric data format.

Maximum Number of Dependent Entities (A22, Max No Dep Ent)

The greatest number of entities that are allowed to be
dependent in any instance of a relation class. Most
relation classes have no exact maximum. For these, "many"
is used to indicate that any number of entities may be
dependent.

Maximum Number of Independent Entities (A127, Max No Ind Ent)

The greatest number of entities that are allowed to be
independent in any instance of a relation class. In a
specific relation class this maximum is always one (1). In
a nonspecific relation class this may be zero or a number
greater than one. Most nonspecific relation classes have
no exact maximum. For these, "many" is used to indicate
that any number of entities may participate.

Minimum Number of Dependent Entities (A21, Min No Dep Ent)

The least number of entities that are allowed to be
dependent in any instance of a relation class.

Minimum Number of Independent Entities (A128, Min No Dep Ent)

The least number of entities that are allowed to be
independent in any instance of a relation class.

Model Object Name (Model Obj Name)

Role name for Object Name; introduced in the Model entity
class.

Model Object Number (Model Obj No)

Role name for Object Number; introduced in the Model entity class.

Module Identification (A129, Module Id)

A means of uniquely identifying a software module.

Null Value (A203)

The value that a DMS places in data fields when no values are supplied by users.

Number of Data Field Occurrences (A191, No of DF Occurs)

The number of times a data field appears in a record type.

Object Creation Date (A132, Obj Creation Date)

The date when a generic object was first recorded in the CDM.

Object Modification Date (A133, Obj Mod Date)

The date when a generic object was last modified.

Object Name (A130, Obj Name)

A word or phrase by which a generic object is known. An object name is either the official name or an alias for an object.

Object Name Type Code (A190, Obj Name Type Code)

A code that indicates whether an object name is an official name or an alias.
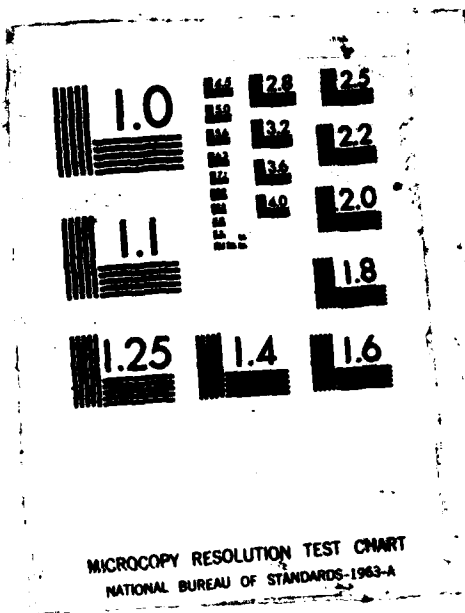
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Object Number (A179, Obj No)

    A number that uniquely identifies a generic object.

Parameter Identification (A31, Parm Id)

    A code that distinguishes a module parameter from the
others for the same software module.

Parameter Sequence Number (A202, Parm Seq No)

    The ordinal position of a module parameter within the list
of parameters for a software module.

Parameter Use Code (A138, Parm Use Code)

    A code that indicates whether a module parameter is used to
supply variable or constant values to a software module or
to return resultant values or error codes.

Preference Number (A140, Pref No)

    A number that indicates the relative acceptability of the
values stored in one data field to those stored in another,
both of which correspond to the same attribute use class.
Preference numbers will be used by the CDMP to determine
which data field to retrieve values from when multiple
choices are available.

PSB Name (A141)

    A name that uniquely identifies a program specification
block.

PSB Sequence Number (A142, PSB Seq No)

    The ordinal position of a PCB within the list of PCBs that
make up a PSB.

Record Key Code (A81, Rec Key Code)

A code that indicates whether the values in a data field can be used to locate instances of a record type and, if they can, whether they must be unique.

Record Set Object Name (Set Obj Name)

Role name for Object Name; introduced in the Record Set entity class.

Record Set Object Number (Set Obj No)

Role name for Object Number; introduced in the Record Set entity class.

Record Type Object Name (RT Obj Name)

Role name for Object Name; introduced in the Record Type entity class.

Record Type Object Number (RT Obj No)

Role name for Object Number; introduced in the Record Tyentity class.

Redefined Data Field Object Name (Redef'd DF Obj Name)

Role name for Data Field Object Name; introduced in the Data Field Redefinition entity class.

Redefining Data Field Object Name (Redef'g DF Obj Name)

Role name for Data Field Object Name; introduced in the Data Field Redefinition entity class.

Relation Class Object Name (RC Obj Name)

Role name for Object Name; introduced in the Relation Class entity class.

Relation Class Object Number (RC Obj No)

Role name for Object Number; introduced in the Relation Class entity class.

Repeating Data Field Index Indicator (A197, Rep DF Index Ind)

A code that indicates whether an attribute use class maps to the index for a repeating data field or to the data field itself.

Repeating Data Field Object Name (Rep DF Obj Name)

Role name for Data Field Object Name; introduced in the Repeating Data Field Occurrence Counter entity class.

Required Membership Indicator (A82, Req Mbr Ind)

A code that indicates whether instances of the member record type in a record set in a CODASYL database must be related to instances of the owner record type in order to exist. If they must be, the DBMS automatically deletes the member instances when an owner instance is deleted and does not permit a new member instance to be created without being re- lated to an owner instance. If a member instance does not have to be related to an owner instance in order to exist, the DBMS does not automatically delete member instances and does permit new member instances to be created without being related to owner instances.

Round/Truncate Code (A180, R/T Code)

A code in a numeric data format that indicates whether a value whose decimal portion is longer than the format's decimal length should be rounded or truncated to fit.

Schema Name (A147)

The name by which the schema of a CODASYL database is known to its DBMS.

Segment Size (A148)

The total storage length of a record type in an IMS data-base.

Segment Starting Byte (A149, Seg Start Byte)

A number that indicates where a data field starts within a record type in an IMS database.

Software Language Name (A25, Lang Name)

The name of the software language in which a software module is written. Some examples are COBOL, FORTRAN, and PASCAL.

Specific Value (A181)

A value of an elementary domain.

Statement Action (A182, Stmt Action)

An NDML action (SELECT, INSERT, MODIFY, or DELETE) that is specified in the ON clause of a constraint statement.

Statement Line Number (A183, Stmt Line No)

The ordinal position of a constraint statement line within those for a constraint statement.

Statement Object Name (Stmt Obj Name)

Role name for Object Name; introduced in the Constraint Statement entity class.

Statement Object Number (Stmt Obj No)

Role name for Object Number; introduced in the Constraint Statement entity class.

Statement Text (A184, Stmt Text)

A fixed-length portion of a constraint statement.

Statement Type Code (A192, Stmt Type Code)

A code that indicates whether a constraint statement is an assertion, a trigger, or a horizontal partition.

Storage Type Code (A159)

A code in a numeric data format that indicates how values are stored. The types are:

Character	(DISPLAY)
Binary	(COMP)
Standard floating point	(COMP-1)
Extended floating point	(COMP-2)
Packed	(COMP-3)

Subschema Name (A151)

The name by which a subschema of a CODASYL database is known to its DBMS.

Total Number of Members (A152, Total Num Members)

A number that indicates how many record types are members in a record set.

To UOM Code

Role name for UOM Code; introduced in the Unit of Measure Conversion entity class.

UOM Code (A185)

A code that uniquely identifies a unit of measure.

UOM Constant Value (A186)

A number in a unit of measure conversion step that is the same every time the conversion is performed.

UOM Name (A187)

A noun or noun phrase that identifies a unit of measure.

UOM Step Number (A188, UOM Step No)

The ordinal position of a unit of measure conversion step within those for a unit of measure conversion.

UOM Step Operator (A189, UOM Step Op)

A symbol that specifies which arithmetic operation (add, subtract, multiply, or divide) is involved in a unit of measure conversion step.

User AP Module Identification (User AP Mod Id)

Role name for Module Identification; introduced in the User Application Process entity class.

User View Object Name (View Obj Name)

Role name for Object Name; introduced in the User View entity class.

User View Object Number (View Obj No)

Role name for Object Number; introduced in the User View entity class.

5.2 CDM1 Attribute Class Glossary Index by Attribute Class Number

AC No	Attribute Class Name
3	Entity Class Label
9	Key Class Number
12	Attribute Use Class Number
14	Attribute Use Class Label
21	Minimum Number of Dependent Entities
22	Maximum Number of Dependent Entities
25	Software Language Name
30	Latest Usage Date
31	Parameter Identification
58	DMS Name
59	DMS Type Code
60	Database Password
75	Database Area Identification
76	Data Type Code
81	Record Key Code
82	Required Membership Indicator
108	Equivalent Attribute Use Class Value
110	Author Identification
112	EC-UV Join Type Code
113	EC-RT Join Type Code
116	Description Line Number
117	Description Text
118	Description Type Code
121	Host Identification
122	DBMS-Accessible Data Field Indicator
123	Key Feedback Length
124	Keyword
125	Linkage Type Code
127	Maximum Number of Independent Entities
128	Minimum Number of Independent Entities
129	Module Identification
130	Object Name
132	Object Creation Date
133	Object Modification Date
138	Parameter Use Code
140	Preference Number
141	PSB Name
142	PSB Sequence Number
147	Schema Name
148	Segment Size
149	Segment Starting Byte

AC No	Attribute Class Name
151	Subschema Name
152	Total Number of Members
159	Storage Type Code
160	Alphabetic Case Code
161	Attribute Class Label
162	Beginning Value
163	Data Field Sequence Number
164	Component Domain Sequence Number
165	Component UOM Sequence Number
166	Data Type Name
167	Decimal Length
168	Description Type Name
169	EC-RT Union Value
170	Ending Value
172	Fill Character
174	Format Number
175	Integer Length
176	Library Identification
177	Library Type Code
178	Maximum Length
179	Object Number
180	Round/Truncate Code
181	Specific Value
182	Statement Action
183	Statement Line Number
184	Statement Text
185	UOM Code
186	UOM Constant Value
187	UOM Name
188	UOM Step Number
189	UOM Step Operator
190	Object Name Type Code
191	Number of Data Field Occurrences
192	Statement Type Code
193	Blocking Factor
194	Device Type Code
195	File/Database Reference Type Code
197	Repeating Data Field Index Indicator
198	CS-IS Mapping Algorithm Use Code
199	CS-IS Mapping Constant Use Code
200	CS-IS Mapping Constant Value
201	Horizontal Parititon Number
202	Parameter Sequence Number
203	Null Value

5.3 CDM1 Attribute Class Glossary Index by Attribute Class Label

Note:

Only attribute classes whose labels are different than their names are listed here. If an attribute class label is not included, it is the same as the attribute class name.

AC Label	Attribute Class Name
AC Label	Attribute Class Label
AC Obj No	Attribute Class Object Number
Alpha Case Code	Alphabetic Case Code
Area Id	Database Area Identification
AUC Label	Attribute Use Class Label
AUC No	Attribute Use Class Number
AUC Obj Name	Attribute Use Class Object Name
Author Id	Author Identification
Begin Value	Beginning Value
Comp DF Obj Name	Component Data Field Object Name
Comp Dom Seq No	Component Domain Sequence Number
Comp UOM Seq No	Component UOM Sequence Number
CS-IS Alg Use Code	CS-IS Mapping Algorithm Use Code
CS-IS Constant Use Code	CS-IS Mapping Constant Use Code
CS-IS Constant Value	CS-IS Mapping Constant Value
Ctr DF Obj Name	Counter Data Field Object Name
DBMS-Acc DF Ind	DBMS-Accessible Data Field Indicator
DB Password	Database Password
DD Obj Name	Data Description Object Name
DD Obj No	Data Description Object Number
Dec Len	Decimal Length
Dep EC Obj No	Dependent Entity Class Object Number
Desc Line No	Description Line Number
Desc Text	Description Text
Desc Type Code	Description Type Code
Desc Type Name	Description Type Name
DF Obj Name	Data Field Object Name
DF Obj No	Data Field Object Number
DF Seq No	Data Field Sequence Number
DI Obj Name	Data Item Object Name
DI Obj No	Data Item Object Number
Dom Obj Name	Domain Object Name
Dom Obj No	Domain Object Number
DT Code	Data Type Code
DT Name	Data Type Name

AC Label	Attribute Class Name
EC Label	Entity Class Label
EC Obj No	Entity Class Object Number
EC-RT Union Discrim Value	EC-RT Union Discriminator Value
Elem Dom Obj No	Elementary Domain Object Number
Elem UOM Code	Elementary UOM Code
End Value	Ending Value
Equiv AUC Value	Equivalent Attribute Use Class Value
F/DB Obj Name	File/Database Object Name
F/DB Obj No	File/Database Object Number
F/DB Ref Type Code	File/Database Reference Type Code
Fill Char	Fill Character
Format No	Format Number
Generated Mod Id	Generated Module Identification
Generating Mod Id	Generating Module Identification
Group DF Obj Name	Group Data Field Object Name
Group Dom Obj No	Group Domain Object Number
Host Id	Host Identification
HP No	Horizontal Partition Number
Ind EC Obj No	Independent Entity Class Object Number
Int Len	Integer Length
KC Mbr AUC No	KC Member Attribute Use Class Number
KC No	Key Class Number
Key Feedback Len	Key Feedback Length
Lang Name	Software Language Name
Lib Id	Library Identification
Lib Type Code	Library Type Code
Linkage Type	Linkage Type Code
Max Len	Maximum Length
Max No Dep Ent	Maximum Number of Dependent Entities
Max No Ind Ent	Maximum Number of Independent Entities
Min No Dep Ent	Minimum Number of Dependent Entities
Min No Ind Ent	Minimum Number of Independent Entities
Model Obj Name	Model Object Name
Model Obj No	Model Object Number
Module Id	Module Identification
No of DF Occurs	Number of Data Field Occurrences
Obj Creation Date	Object Creation Date
Obj Mod Date	Object Modification Date
Obj Name	Object Name
Obj Name Type Code	Object Name Type Code
Obj No	Object Number
Parm Id	Parameter Identification
Parm Seq No	Parameter Sequence Number

AC Label	Attribute Class Name
Parm Use Code	Parameter Use Code
Pref No	Preference Number
PSB Seq No	PSB Sequence Number
RC Obj Name	Relation Class Object Name
RC Obj No	Relation Class Object Number
Rec Key Code	Record Key Code
Redef'd DF Obj Name	Redefined Data Field Object Name
Redef'g DF Obj Name	Redefining Data Field Object Name
Rep DF Index Ind	Repeating Data Field Index Indicator
Rep DF Obj Name	Repeating Data Field Object Name
Req Mbr Ind	Required Membership Indicator
R/T Code	Round/Truncate Code
RT Obj Name	Record Type Object Name
RT Obj No	Record Type Object Number
Seg Start Byte	Segment Starting Byte
Set Obj Name	Record Set Object Name
Set Obj No	Record Set Object Number
Stmt Action	Statement Action
Stmt Line No	Statement Line Number
Stmt Obj Name	Statement Object Name
Stmt Obj No	Statement Object Number
Stmt Text	Statement Text
Stmt Type Code	Statement Type Code
Total Num Members	Total Number of Members
UOM Step No	UOM Step Number
UOM Step Op	UOM Step Operator
User AP Mod Id	User AP Module Identification
View Obj Name	User View Object Name
View Obj No	User View Object Number

END

7-87

DTIC